Reference Manual

IBM 7750 Programmed Transmission Control

Programming Logic and Organization

IBM

IBM

Reference Manual

IBM 7750 Programmed Transmission Control

Programming Logic and Organization

CONTENTS

LIST OF FIGURES

INTRODUCTION

A "TELE-PROCESSING®" System can be described as a combination of products, components,and communication links permitting the processing of data at a location remote from its point of origin. Generally, such systems are designed to function as an in-line/on-line operation, processing data as it originates and updating all records affected. It is, however, not limited to in-line operation since batch or schedule type data are also handled. Such a system is considered to be an operational control system which provides direction to an enterprise, or control over several applications within an enterprise, by furnishing information where and when it is needed upon computer control or operator intervention over communication links.

A "TELE-PROCESSING" System can be thought of as a Central Processor (computer) connected to a number of scattered input-output devices, known as Terminals, through a number of simultaneously operating Communication Channels. The Central Processor also has the normal input-output units such as card readers, magnetic tape units, disk files, printers,and punches as required.

The advent of "TELE-PROCESSING" has introduced a number of new concepts and additional functions which are not familiar to conventional data processing operations. The complex represented by TERMINALS, COMMUNICATION CHANNELS, and multiplexing devices (if any) is called the NETWORK of such a system. The Network presents such new concepts as: numerous remotely located input-output units called Terminals,which are connected to Communication Channels; the type of Terminal may vary and the speed of transmission (bit rate) over the Communication Channels may vary; many Terminals (each on a different Communication Channel) may be simultaneously sending and/or receiving data; various transmission codes may be used within one system, etc.

In order to efficiently link the Network to the Central Processor, a new machine, the IBM 7750 Programmed Transmission Control, was developed. This new machine was designed to recognize these new concepts and perform these new functions in such a manner that the Central Processor, instead of having to deal with the individual elements of this varying complex described above, now will be dealing with only one unit,which appears to it as simply another input-output device.

Figure 1 is a schematic representation of the IBM 7750 and shows its functional location in a "TELE-PROCESSING" system, between the Network and the Central Processor. The IBM 7750 Programmed Transmission Control consists of five so-called Racks:

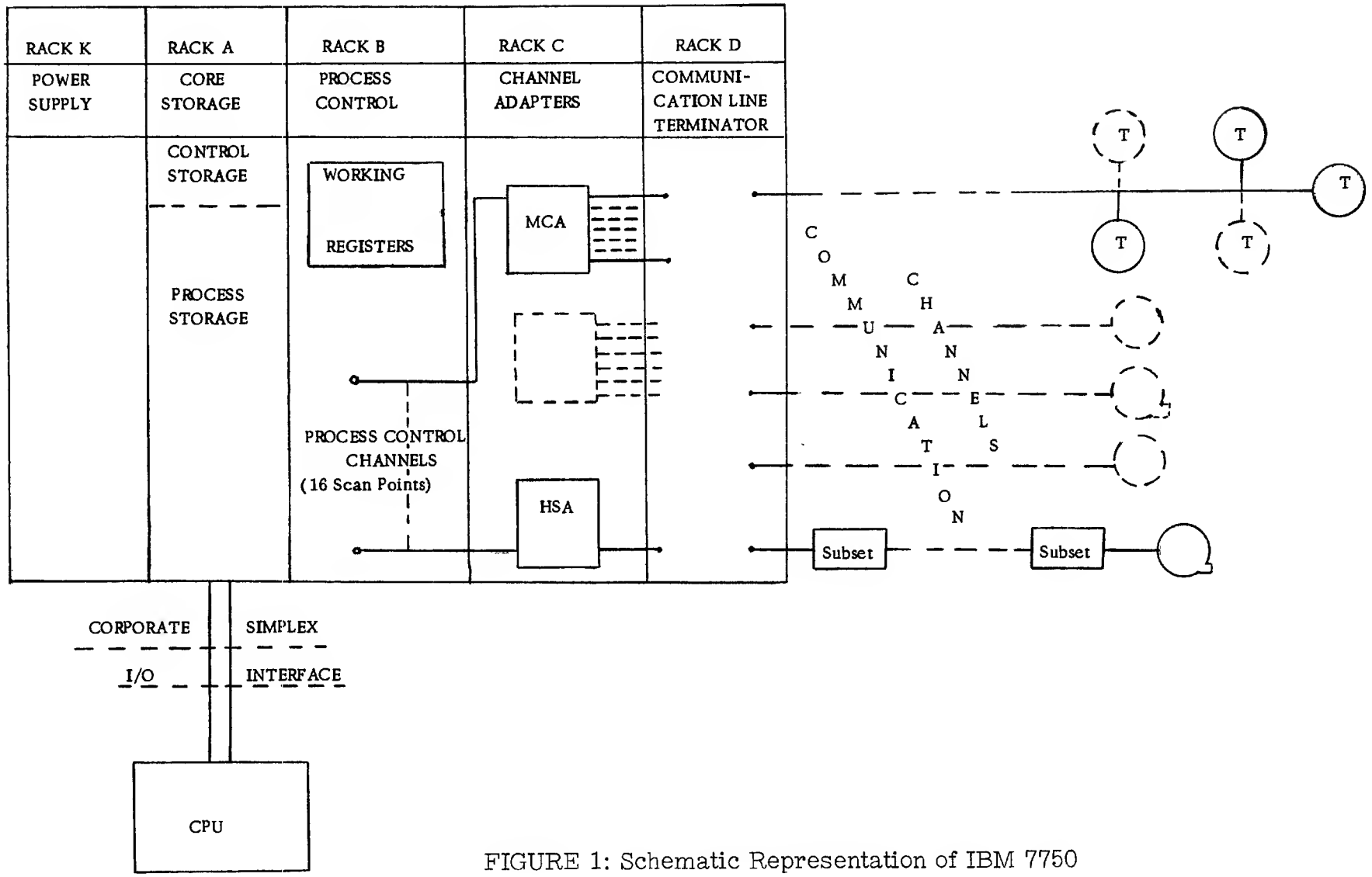


FIGURE 1: Schematic Representation of IBM 7750

RACK D - Communication Line Terminator; The Network Communication Channels are connected to the IBM 7750 in this Rack.

RACK C - Channel Adapters; they perform a variety of functions, such as: change voltage levels from those used in Communication Channels to those used in IBM circuits and vice versa, control digital subsets, and time multiplex a number of low-speed Communication Channels with one or more high-speed channels; only certain configurations of low and high-speed channel adapters are possible.

RACK B - Process Control; this Rack provides the logic and circuits for performing three basic functions: 1) Assembly and distribution of data between 16 scan points and Process Storage; 2) Processing of assembled data under the control of stored programs; 3) Transmission of data from and to the Central Processor via the Simplex I/O Interface.

RACK A - Core Storage; this consists of 4K, 8K,or 16K - 48 bit words of Process Storage, and 128 - 48 bit Control Storage words.

RACK K - Power Supply

The Processor receives information units, called messages, from numerous remote and local Terminals through the IBM 7750, performs the necessary processing, updates all records affected, and then furnishes the information required to the originating and/or other Terminals of the Network through the IBM 7750 when and as needed. On the other hand, a message transmission or exchange may be initiated by the computer.

The IBM 7750 is itself an electronic computer, performing automatic and programmed functions. It is a binary machine. Automatic functions are performed when certain conditions are recognized in specific hardware components (registers, triggers, etc.) by wired-in machine logic. Programmed functions are performed when the logic recognizes specific commands given in predetermined binary bit patterns known as Operation Codes.

The automatic functions of the IBM 7750 are:

1.) Assembly (input) and distribution(output) of messages; these functions are performed on a character-by-character basis.

2.) Transfer of data to and from the Processor.

3.) Detection and indication of certain Communication Channel and Process Control Errors.

The programmed functions are:

1.) Preparation of input data for the Processor and output data for the Terminals; code conversion, editing (i. e., deleting and inserting functional characters), and many other functions such as validity check, count-keeping, message numbering,and, in some cases, even format changes.

2.) Control over the Input-Output operation of the entire Network including the maintenance of the status of awareness.

The IBM 7750 has a parallel operational system: while the Input-Output operations progress automatically, the execution of the stored program and the data transfer to and from the Processor are also performed.

The conditions imposed by the Network operation, the specific requirements of the input-output formats, and the real-time time schedule obviously result in system constraints which must be adhered to when designing the program for the Central Processor.

These constraints exert an even greater influence on the IBM 7750 programs, since a unique approach is necessary in order to meet the requirements of such a system.

This manual may serve as a guide for Systems and Programming personnel in the design and organization of their operational specifications and programs. Hardware functions and programming techniques are described in adequate detail for this purpose. The manual provides a ready reference for the selection of the most suitable instructions for each programmed function. The Instruction Set is introduced in convenient table-formats which are preceded by a descriptive index and an explanation of the symbols used in the mnemonic Operation Codes and in the description of the individual instructions. For further convenience, a serial number is assigned to each instruction.

The organization and presentation of the Subject Matter in this manual was selected and designed so as to best facilitate its use as a reference guide in conjunction with the IBM 7750 Reference Manual, when published. The INTRODUCTION TO THE IBM 7750 manual (revised edition is available) discusses the general organization, functions, operational methods,and other characteristics of the machine, and familiarity with its contents is presumed.

SECTION I

OPERATIONAL PRINCIPLES

The prime function of the IBM 7750 is to service a number of Communication Channels simultaneously; that is, to give service to each of the channels in any phase of the operational procedure without delay. The Communication Channels are automatically scanned by the Process Control Scanner through the Process Control Channels and the Channel Adapters on a predetermined schedule. The maximum number of Process Control Channels on an IBM 7750 is 16; each Process Control Channel is connected to a Scanning Point. In order to maintain a well balanced constant flow of data, the program written for controlling the Input-Output operations and the associated data preparation must be well-coordinated with the hardware functions, thereby providing a means for simultaneous input-output operation. The Program must work on a "scan basis", examining the status of the existing conditions for each Communication Channel and for the Processor at a given moment, and executing the instructions necessary to satisfy the detected conditions. During the program design, core memory areas are assigned, by means of Limit Words, to each Communication Channel, and the Processor (see Section V-6).

The programmed "scanning" occurs at certain time intervals, the length of which is variable. An interval depends upon the number of memory areas to be"scanned"and the time necessary to execute the required instructions, if any, which provide service according to the status at the time any particular area is "scanned". If the "scanning" schedule of the different core memory areas and indicators is well designed within the program, these time intervals will remain reasonably proportional and thus, in effect, provide simultaneous service to all components.

A simplified example in support of these statements would be to consider the processing of input data as an independent procedure without interruption by any other necessary operations (such as polling, data transfer to and from the computer, processing, and queueing output messages, etc.) that obviously must be implemented in an actual program. The IBM 7750 processes data character by character. Each input-data-producing channel has an exclusive input area, accessible by the program through a unique Limit Word, in which the incoming data is stored character by character. The program does not delay the processing until a complete message is stored in an input area. It goes from area to area in a predetermined sequence and processes the input characters in each. Finally, it returns to the first area to process, or attempt to process (another character may not have been stored there yet), the new characters stored there since the program left it. At this point, the program begins its next go around in order to process the recently stored characters in each of the input areas. Since the processed characters in the various input areas most probably represent

a different part of a message, they may require different program services. This is true even if the processing of uniform messages is assumed (i. e., fixed length messages of the same size, having contents of the same nature in identical sequence). Consequently, as the program proceeds from area to area, and from character to character within each area, different routines may be required. Since it is impossible to know or predict which particular routines will be required, the program must be so organized that all necessary routines are available to all areas at all times.

SECTION II

GENERAL PROGRAM ORGANIZATION PRINCIPLES

The IBM 7750 has six operation modes. These modes, in decreasing priority order are: Service, Channel Service, Copy, Out, In, and Normal Modes. The contents of two registers, the Mode Status Register (MSR) and the Mode Request Register (MRR), determine in which mode the machine is operating and the change from one mode to another. The program has access to the Mode Request Register only. In some cases, the preparation for a mode change is an automatic machine function, whereas in other cases it must be programmed by setting, or resetting, the proper bit in the MRR. In each case, the recognition of the highest requested priority, or the lack of request for priority, is automatic; therefore, the method is called Automatic Priority Processing. (See Mode Selector Operation in Section VII.)

With the exception of the Copy Mode, a program must be written for each mode. These are called Mode Programs. The programs of priority modes are intended to satisfy special conditions and should, therefore, perform specific functions. Nevertheless, the programmer is not entirely restricted to follow binding rules in designing these programs. If the programmer can devise a safe and efficient means to do so, he may take advantage of the higher priority rank of some modes to perform important functions other than those for which the modes were designed. On the other hand, in order to shorten the execution time of some priority (Out, In) mode programs, he may incorporate a limited amount of preparatory work into his Normal Mode Program. This latter case is somewhat more desirable, but in both cases, very good judgement must be used to avoid unnecessary complications and inefficient results.

The Normal Mode Program is the main program of the IBM 7750. Its purpose is to perform the processing functions required for each application (polling, routing, code-conversion, editing, keeping counts, generating and checking Check Characters, message numbering, removing and returning empty blocks, etc.). Because of the IBM 7750's unique processing approach, the Normal Mode Program must function similar to a "control program" providing numerous independent routines which are integral parts of the entire complex. The same routines must be available to each Communication Channel's core memory area, as required.

Since the recognition of the highest requested priority (or no priority request) is automatic, one cannot predict when the execution of each of the Mode Programs may become necessary. Similarly, program routines within a Mode Program - and this applies to any Mode Program - may be called for at any time, depending on certain conditions found by the program. In conclusion, it must be stated that all Mode Programs and their routines must be retained in Process Storage throughout the operation of the System. This applies also to the tables, constants, and

Limit Words these programs use. Only "emergency" programs (i.e. programs to be executed only in case one of the system components fails) or "diagnostic" programs (used for maintenance) can be stored outside the core memory and called for when required.

In designing the Mode Programs, two most important objectives should be kept in mind:

1. the amount of core memory area occupied by programs, tables, constants, and Limit Words must be kept as low as possible, and at the same time the buffer storage area should be increased as much as possible.

2. in order to provide the fastest possible service to all Communication Channels, the execution time of each Mode Program and each routine within these programs must be minimized without sacrificing accuracy or safety.

Only programs of maximum efficiency can meet these requirements.

SECTION III

CORE MEMORY WORDS AND THEIR FUNCTIONS

The IBM 7750 is a stored program computer with two core memories: Control Storage (CS) and Process Storage (PS). Control Storage consists of 128 words, and Process Storage may have 4096, 8192, or 16,384. Control Storage contains the words necessary for controlling the mode operations and the Input-Output operations as well as words required for the indication of certain automatically detected channel errors. Process Storage houses the programs, constants, tables, area (chain) controlling Limit Words, and data.

1. CONTROL STORAGE WORDS

Depending on the network configuration, Control Storage may hold 2 or 3 types of functionally defined words and some unassigned words, as follows:

A. PROCESS WORDS (PWD)

There are six Process Words with predetermined locations. Each controls the machine operation in a specific mode. With the exception of the Copy Mode Process Word, each contains a 14 bit Instruction Counter (IC), three 11 bit registers called X, Y, and Z Registers, and a Word Parity Bit. Z is a count-down register; its content (which is treated as an 11 bit binary number) may be decremented by one. When executing Address and Limit Moving instructions, the 16 bits composed of Register Y and 5 low-order bit positions of Register X comprise the Operational Address Register (OAR) and hold an Address or a Limit. At such times the machine logic interprets these16 bits as one 16 bit register. The 11 high-order bits of the Address or Limit are stored in Register Y and the 5 low-order bits in Register X. However, when an Address is moved from OAR into a PS word, the contents of OAR will be rearranged and stored in proper sequence in the Address field of the Process Storage word addressed by the instruction. Likewise, when an Address is moved into OAR, the bits are rearranged into the OAR-required reversed sequence by the machine logic. A Limit in a Process Storage word has the same arrangement as the OAR. The Copy Mode operation is a fully automatic procedure requiring no program; therefore, its Process Word, called the COPY WORD, serves a different purpose and has a different format corresponding to it. It contains a 16 bit Address field (Start Address), a 16 bit Limit field (Stop Address), an 11 bit Z Register, and a Word Parity Bit. (See Appendix I, Charts 1, 2, and 3 for PWD formats).

The execution of the instructions is sequential; the Instruction Counter is automatically incremented by one during the execution of each instruction. Each of the five programmed modes has its own Instruction Counter in its own Process Word. Whenever a recognized mode request initiates a mode change, the IBM 7750 switches to the new operation mode by accessing its Process Word. The change is performed automatically through wired in circuitry according to the status of the updated Mode Status Register. When the Process Word of a programmed mode is accessed,. the Instruction Counter contains the address of the next sequential instruction following the one already executed before a mode change temporarily terminated the operation of that particular mode. This is so because the Instruction Counter was incremented by one during the execution of the instruction prior to the mode change.

As an example, assume that the program of one of the higher priority modes is being executed. Obviously, the very last instruction of any Priority Mode Program must be a Branch instruction, back to the first instruction of the same program. However, in order to leave that mode, the program must reset the proper bit in the Mode Request Register. The instruction which resets this bit will be the one preceding the Branch instruction. As soon as the reset instruction has been executed, the Instruction Counter is incremented by one and now contains the address of the next instruction,which is the Branch instruction. As at the end of the execution of every instruction, the Mode Status Register will be compared with the Mode Request Register. Assuming, for this example, that no request bit for the same or higher priority mode is set in the MRR before the compare is completed, the machine will switch to the next lower ranking mode. When the just-left mode is again requested as highest in priority among the requested modes, the first instruction to be executed in this mode will be the Branch instruction, since its address is located in the Instruction Counter when the Process Word is accessed.

B. CHANNEL WORDS (CWD)

Each high or low-speed Communication Channel connected directly to the IBM 7750 must have a unique control word assigned to it for the assembly or distribution of characters received or sent on that particular Communication Channel. For half-duplex Communication Channels one control word is used alternately for assembly and distribution. For full-duplex Communication Channels two control words are required, one for the incoming and the other for the

outgoing line. The expression, "directly connected Communication Channels," refers to every Communication Channel (whether from remote or local Terminals) connected to the IBM 7750 through some type of Channel Adapter. In some systems, there are special multiplexing devices, called Exchanges or Interchanges, located in remote geographical places, for receiving data from a number of Terminals on low-speed channels and forwarding them to the IBM 7750 on a (usually full-duplex) high-speed channel,and vice versa. In such cases, only the high-speed channels have Channel Words assigned in Control Storage.

Channel Words have predetermined locations in Control Storage. Each Channel Word contains a 16 bit Address field, an 11 bit Assembly Area, a 20 bit Control Area, and a Word Parity Bit. The Address field holds the Address of an 11 bit character field in Process Storage, where the assembled character will be stored (Receive Status) or from which the next character will be obtained (Send Status). In Receive Status, the bits are assembled into a character in the Assembly Area. In Send Status, a character is transmitted out bit-by-bit from the Assembly Area. The Control Area may be set different ways, depending on the type of the transmitted characters and the operational status of the Communication Channel. These subjects are described in detail in Section IV-3-C. (See Appendix I, Chart 4 for general CWD format).

C. ERROR CHANNEL WORDS (ECWD)

Two Error Channel Words with predetermined locations in Control Storage are assigned to each half-duplex and full-duplex high-speed Communication Channel. Consequently, a half-duplex high-speed channel has assigned to it one regular Channel Word and two Error Channel Words. A full-duplex high-speed channel has assigned to it two regular Channel Words and two Error Channel Words.

Each Error Channel Word has not only a predetermined Control Storage location, but it is associated with a specific high-speed channel error. Therefore, when an error occurs,which is indicated to the program through an ECWD, the Control Storage address of the ECWD identifies not only the channel, but reveals the type of error detected on the channel. Although high-speed channels transmit Synchronous type characters, Start-Stop type Error Channel Words are used to detect certain high-speed Channel Errors in order to provide more favorable conditions for the machine logic. Therefore, the program must set bit 14 to Zero in all Error Channel Words to indicate the Start-Stop status. Another requirement is that the Character Length (bits 30-27) must be set to a number higher than 11_{10}

(i. e. higher than the maximum valid character length), which may be any number between 12_{10} and 15_{10}. The detection of this invalid Character Length value by the hardware will prevent the machine from performing bit operation in bit positions 11-1, when such a word is located in the Control Storage Data Register. The Not Hold Bit (bit 26) and the Not Error Bit (bit 13) must be set to logical Ones. The Send-Receive Bit (bit 12) of a Time Out Tag ECWD assigned to a half-duplex high speed channel must be set to logical One. In all other cases, the Send-Receive Bit must be set to logical Zero. When, following an error indication, the operation of the channel is restored to normal, the four low order bit positions of the Character Control field (bits 18 - 15) must be loaded with a special bit pattern (14_8) in order to clear the error condition in the Channel Adapter. Therefore, these bits must have initially been set to Zeros and may not be used by the program, as long as the operation of the channel is normal. The rest of the bit positions in the Error Channel Word are for the optional use of the program since the machine logic never examines their contents. Cross-references, such as the Control Storage address of the corresponding regular Channel Word(s), the program-assigned channel identification number(s) or other reference items, needed by the program, may be kept in these bit positions.

The use of the Error Channel Words is described in detail in Section XII-1.

The format of the Error Channel Word is shown in Appendix I, Chart 5.

D. SCRATCH WORDS (SWD)

Scratch Words are functionally unassigned words with predetermined locations in Control Storage. Two Scratch Words are available and they may be used at the discretion of the programmer.

Since the total number of functionally assigned words (Process Words, Channel Words, Error Channel Words) plus the two Scratch Words may not exhaust the capacity of Control Storage, the remaining unassigned words could be used as additional Scratch Words by the programmer. However, since in any system, additional Communication Channels may be added at any time, it is not advisable to utilize these available words in this manner in order to avoid unnecessary program changes and relocations in the future.

The program cannot change functional word assignments in Control Storage or relocate any functionally assigned word, since they have predetermined locations for wired-in recognition.

2. PROCESS STORAGE WORDS

Process Storage holds 3 types of words. None have predetermined locations and functions. Their functions and, consequently, their formats are determined by the source program. Their locations are determined by the source and/or the Assembly Program.

A. DATA WORDS (DWD)

A Data Word contains four 11 bit fields for 4 data characters, 3 unused bit positions (36-34) and a Word Parity Bit. The character fields are identified as the A (bits 47-37), B (bits 33-23), C(bits 22-12) and D (bits 11-1) fields. (See Appendix II, Chart 1).

B. LIMIT WORDS (LWD)

The Limit Word is primarily designed as a control word for a chain; however, the programmer may find several other useful applications for this word format. The LWD has a 16 bit Address field, a 16 bit Limit field, an 11 bit D character field, four unused bit positions, and a Word Parity Bit. The functions and most frequent assignments for the LWD format are described in Section XI-3. (See Appendix II, Chart 2 for LWD format).

C. INSTRUCTION WORDS (IWD)

Each instruction is stored in a full word. It has a 16 bit Address field (W) to hold the Process Storage operational address, two 3-bit fields for naming registers for data manipulation (R) and address modification (M), two 4-bit fields for specifying field sizes in connection with R and M registers (S and L respectively), two one-bit-position Flag fields for indicating Prevent Mode Change and Decrement Z Register commands, a 10 bit Operation Code field, an Op-Code Parity Bit associated with the Operation Code and the two Flag Bits, a Word Parity Bit, and four unassigned bit positions. As will be shown later, R and S may also be interpreted as a combined 7-bit field which is used to define a Control Storage address in SS type instructions. The IWD is the only type of word which has two parity bits. All other types of words, either in Control Storage or in Process Storage have only one parity bit, the Word Parity Bit. The IBM 7750 recognizes odd parity as the valid condition. (See Appendix II, Chart 3 for IWD format.)

SECTION IV

INPUT-OUTPUT OPERATIONS

All references to Input-Output operations in this manual refer to the transmission of messages between the IBM 7750 and the Terminals through Communication Channels. Although the transmission of data between the Central Processor and the 7750 is a type of Input-Output operation, it will not be referred to as such. This will be called data transfer.

1. CHARACTER ASSEMBLY AND DISTRIBUTION

Character assembly and distribution is an automatic procedure, and when such an operation is performed, the affected Channel Word will be found in the Control Storage Data Register. The affected Channel Word is accessed through the address provided by the Channel Adapter after the latter has been selected by the Process Control Channel Scanner.

The Input-Output procedure is performed on a single character by character basis. Bits received on a Communication Channel are assembled into a character in the Assembly Area of the unique Channel Word permanently assigned to that channel. When a complete character is assembled, it is stored in Process Storage. The 16 bit Address in the Channel Word provides the address of an 11 bit character field in Process Storage where the assembled character is to be stored. After a character has been stored in Process Storage, it is located in the low-order portion of the addressed character field. The remaining high-order bit positions contain Zeros. Similarly, when sending a message from the IBM 7750, a character specified by the 16 bit Address in the Channel Word is obtained from Process Storage, placed in the Assembly Area and then transmitted to the channel bit-by-bit. When obtaining a character from Process Storage, all 11 bits of the addressed character field are loaded into the Assembly Area regardless of the actual length of the character.

The Assembly Area is an 11 bit shift register. Bits in the Assembly Area are shifted one position towards the low-order bit position each time a bit is received from (Receive Status) or sent to (Send Status) the channel. The shift operation is completely different when the Extended Assembly Area is used in Receive Status while seeking for Character Synchronization. (See Section IV-4, C).

The character address in the Channel Word is incremented by one each time a character has been stored in, or obtained from, Process Storage. When the

five low-order bits of the Address become all Ones (i.e., End of Block condition), the Channel Service Mode bit is automatically turned on in the Mode Request Register, thereby requesting Channel Service. Should this service not be immediately available, a new attempt will be made to obtain the service each time the Channel Word is read into the Control Storage Data Register. As long as the Address in the Channel Word contains all Ones in the five low-order bit positions, no character will be stored in, or obtained from, Process Storage. When the service is available, the Channel Service Mode Program will be executed, resulting in a new Address in the Channel Word for the next character.

Each time a character is to be stored in, or obtained from, Process Storage, a Character Interrupt occurs. Character Interrupt is an automatic interruption of normal machine operation for one machine cycle (28u). This is the time necessary for the machine logic to perform the character-move operation, the Address-increment operation in the Channel Word, and the test for End of Block condition.

2. CHARACTER TYPES

The characters processed by the IBM 7750 can be transmitted in two general ways: START-STOP and SYNCHRONOUS.

A Start-Stop character consists of a Start bit, a certain number of Data bits, and one or more Stop bits. The Start bit is always a logical One; the Data bits may be logical Ones and/or logical Zeros which compose specific characters in predetermined combinations; the Stop bits are always logical Zeros. The Stop bit (or the last Stop bit if there are more than one) may be longer in time interval than an interger bit time. If it is longer, the Stop-bit times are referred to as interger and fractional Stop-bit times.

A Synchronous character consists of all data bits. It may be in any predetermined combination of logical Ones and logical Zeros representing a valid character.

3. THE PREPARATION OF CHANNEL WORDS

A. GENERAL DESCRIPTION

The Network-configuration of a specific system may consist of full-duplex, half-duplex, simplex, or a combination of these types of high and/or low-speed Communication Channels. One or more Terminals may be connected to each of these channels. In some applications, multiplexing devices, called Exchanges or Interchanges, are also connected to the Central Installation to serve as intermediate units, as described in Section III-1, B. The

Input-Output operation on each channel connected to the IBM 7750 from Terminals or intermediate multiplexing devices is controlled by a unique CWD assigned to that channel. One CWD is assigned to a half-duplex channel and two to a full-duplex channel (one for the input and one for the output line).

In order to perform the desired function(s) on a Communication Channel, certain fields in the Control Area (bits 31-12) of the corresponding Channel Word must be set by the program. This manipulation of these fields provides the "communication" between the program and the machine logic. Four different basic settings are possible, depending on the type of characters the transmitted messages consist of,and whether sending or receiving. The machine can perform automatic changes on the basic settings to modify or reverse the operational status of the channel and/or the functional status of a Channel Word. This will happen as the result of the recognition of certain special characters in the Assembly Area of the CWD. These special characters will either be obtained from Process Storage or will be transmitted by Terminals. If obtained from Process Storage, the program must allocate these characters to the proper Process Storage locations. If transmitted by Terminals, the program must prepare the conditions for recognition. In addition, for some cases the program prepares an entire new section for the Control Area of certain Channel Words. These sections are stored in predetermined Process Storage locations and will automatically be accessed and loaded into the Control Area upon the recognition of a special character. In all cases, such changes are prescheduled and arranged by the program; therefore, the programmer has complete control over these operations. They will be described in detail in the following chapters.

B. THE ORGANIZATION OF THE CHANNEL WORD

Regardless of its functional status, each Channel Word is composed of the same basic areas (Appendix I, Chart 4). The main areas of a CWD are:

1.	Address	16 bits (47-32)
2.	Control Area	20 bits (31-12)
3.	Assembly Area	11 bits (11-1)
4.	Word Parity	1 bit (P)

The Address, Assembly Area, and Word Parity have similar functions in each CWD regardless of its status. The Control Area is composed of the same basic fields for all applications; only the functions of certain bits within the Status-Micro, Character Control, and the Status-Macro

fields differ, depending on the transmitted character type and the functional status. The functions of the Address and the Assembly Area have already been described in Chapter 1 of this Section. The following paragraphs describe the fields within the Control Area, and their functions.

LAST TIMING BIT (bit 31) in all Channel Words

Used in Receive and Send Statuses.

This bit is automatically set and constantly checked against the value of the Timing Line in the Adapter Control Interface in order to determine when the channel demands a bit from (Send Status), or is ready to transmit a bit to (Receive Status), the Assembly Area. It is of no concern to the programmer; its basic setting is irrelevant.

CHARACTER LENGTH (bits 30 - 27) in all Channel Words

Used in Receive and Send Statuses.

This field contains the number of bits in a character. The program sets this field. For a Start-Stop character, the Start and Stop bits must be included in the bit count; fractional Stop bits are omitted. The highest valid setting is 11_{10} (13_8).

STATUS MICRO (bits 26 - 23) in Start-Stop Channel Words

a. NOT HOLD BIT (bit 26)

Used in Receive and Send Statuses.

This bit must be set to logical One by the program in order to enable the channel to perform its function. If this bit is reset to logical Zero, the CWD is placed in HOLD Status and the channel becomes inactive relative to the Process Control. In inactive status, all Scan-cycle functions are inhibited for the channel (i. e., no bit operation can take place in the Assembly Area of such a CWD). The bit may be reset by direct programming; however, this is usually used only when the channel is in error. Normally, it is reset automatically by the machine logic due to certain conditions after a Status Change has been established, as described in Section IV-4D. The Status Change must be initiated by the program, but the Hold Status becomes effective only if certain conditions following the Status Change are recognized by the machine logic. In actual machine performance, bit 26 and bit 14 are tested by the hardware in every CWD in order to establish, or maintain, the

active, or inactive, status of the channel. A CWD is in Hold Status and its channel inactive only if both bit position 26 and bit position 14 of the CWD contain logical Zeros. Since bit 14 must be a logical Zero in Start-Stop CWD (specifying the Start-Stop classification of the CWD), the programmer must concentrate on the setting and testing of the Not Hold Bit alone. Dependent upon the described settings, the CWD is referred to be in NOT HOLD STATUS (active) or in HOLD STATUS (inactive).

b. DELAY BIT (bit 25)

Used in Send and Receive Statuses.

This bit is set to logical One automatically by the recognition of a Status Change Character (SCC) or a Sending Delay Character (SDC) in the Assembly Area. The application and function of these characters and the Delay Count Character (DCC) associated with them is described in Section IV-4, D and E. While the Delay Bit is on, no bit transmission takes place from the Assembly Area. The bit is reset to logical Zero automatically when the DCC which follows the SDC is decremented to all Zeros, or when the first bit is received from the channel before the DCC trailing the SCC is decremented to all Zeros. However, in the latter case when the time elapsed (and consequently the DCC is decremented to all Zeros) without receiving a bit from the channel and the CWD is automatically placed into HOLD status, the Delay Bit must be reset by the program.

The program must set this bit to Zero in the basic setting of a Start-Stop Send or Receive CWD.

c. FRACTIONAL SAMPLING BIT (bit 24)

Used in Send Status only.

This bit must be set to logical One by the program if characters having non-integer Stop bits are sent to the channel. This will cause the Channel Adapter to elongate the last Stop bit by a predetermined fraction. Its setting is irrelevant in a Start-Stop Receive CWD.

d. TRANSFER CHECK BIT (bit 23)

Used in Send Status only.

Each time a bit is sent to the Channel Adapter from the low-order bit position of the Assembly Area, it is also shifted into bit position 23 of the CWD. During the next bit time, Process Control compares

the bit sent to the Channel Adapter with the Transfer Check Bit. A difference indicates a channel error, and the Channel Check Light on the Operator's Panel will automatically be turned on. The error is automatically indicated to the program as described in Section XII-1. The Transfer Check Bit is set automatically, therefore, its basic setting is irrelevant.

e. Bits 24 and 23 are not used in Receive Status.

STATUS MICRO (bits 26-23) in Synchronous Channel Words

a. EXTENDED ASSEMBLY AREA (bits 26-23)

Used in Receive Status only when Character Synchronization is required.

The function of this field is explained in Section IV-4, C. Its setting is irrelevant because it is automatically cleared before it is used.

b. BIT 26 in Send Status

The program must set this bit to logical Zero, together with bit 14 in order to inactivate the channel after a Data Transfer Error has been indicated and an error routine is to be executed. (See Section XII-1, E). However, if the programmer is certain that this bit was Zero, either because the CWD is assigned to the Sending line of a full-duplex channel and it had been set to Zero in the basic setting, or the program set this bit to Zero when it changed the status of a half-duplex channel from Receive to Send, the resetting of bit 14 alone will produce the desired Hold Status. There seems to be little reason to inactivate a Synchronous character transmitting channel through its regular CWD for other reasons. In case of other types of channel errors, such a channel will be inactivated through the Error Channel Word corresponding to the indicated error.

c. NOT ERROR BIT (bit 24)

Used in Send Status only.

The program must set this bit to logical One, indicating a "not error" condition. It is automatically set to Zero when a Data Transfer Error is detected on the channel. (See Section XII-1, C.)

d. TRANSFER CHECK BIT (bit 23)

Used in Send Status only.

Its function is the same as in a Start-Stop Send CWD. Its basic setting is irrelevant.

e. Bit 25 is not used in Send Status.

SEQUENCE COUNTER (bits 22-21) in all Channel Words

Used in Receive and Send Statuses. Optional.

If the Action Delay feature is used, this field must be set to a number between 0_{10} and 3_{10} each time the Action Delay Bit is set to logical One. See description in Section IV-4, F. Whenever the Action Delay feature is not required, this field must be set to Zeros.

ACTION DELAY BIT (bit 20) in all Channel Words

Used in Receive and Send Statuses. Optional.

If the Action Delay feature is used, this bit must be set to logical One each time the recognition of the Action Delay Character is required. This will cause the machine logic to seek for the Action Delay Character in the Assembly Area of the CWD. See description in Section IV-4. F. Whenever the Action Delay feature is not required, this bit must be set to Zero.

CHARACTER CONTROL (bits 19-15) in all Channel Words

Used in Receive and Send Statuses.

This field is used in all Channel Words to initiate Adapter Synchronization. This procedure is described in Section IV-3, D.

Since bit position 19 contains a One bit only when Continuous Hunt is chosen for Character Synchronization on Synchronous Receiving channels, this bit is referred to as CONTINUOUS HUNT BIT in Synchronous Receive Channel Words.

The second function of the Character Control field is bit count. As the last step of the Adapter Synchronization, the four low-order bits of this field are automatically reset to Zeros and may be used as BIT COUNTER. The Bit Counter is used when bits of any type of character are sent from, or bits of Synchronous type characters are received in, the Assembly Area of a CWD (when Start-Stop type characters are received, the bits are not counted). Each time a bit is shifted out of the Assembly Area,or

a bit of a Synchronous character is received in the Assembly Area, the value of the Bit Counter is automatically incremented by one; therefore, it always contains the number of bits that have been sent to, or received from, the channel for a character. After the increment, the value of the field is compared with the Character Length. On equal compare, Character Interrupt is initiated; the next output character is obtained from, or the assembled character is stored in, Process Storage; the Bit Counter is automatically reset to Zero.

STATUS MACRO (bits 14-12) in Start-Stop Channel Words

a. START-STOP BIT (bit 14)

Used in Receive and Send Statuses.

This bit defines the type of characters transmitted on the channel. For Start-Stop type characters, it must be set to logical Zero, and it is called the Start-Stop bit. It must be set by the program.

b. NOT ERROR BIT (bit 13)

Used in Receive and Send Statuses.

This bit is used by the machine logic to indicate a Channel Error. When a Channel Error occurs, the machine logic changes its setting from One to Zero. Therefore, the program must set this bit to a logical One in the basic setting to indicate "not error" condition. After an error indication, the program must again set it to One before the operation of the channel is reinstated. (See Section XII-1, E).

c. SEND-RECEIVE BIT (bit 12)

Used in Receive and Send Statuses.

The status of this bit position indicates the direction of data flow on the channel. A One bit in this bit position means Send Status, a Zero bit indicates Receive Status. The program must set this bit (or initiate the change of its setting when the Status Change feature is used) .

STATUS MACRO (bits 14-12) in Synchronous Channel Words

a. SYNCHRONOUS BIT (bit 14)

Used in Receive and Send Statuses.

A logical One in this bit position indicates that Synchronous type characters are transmitted on the channel. It must be set by the program.

b. CONTROL BIT (bit 13)

Used in Receive and Send Statuses.

This bit has functions in detecting Channel Errors. In certain cases, the setting of this bit determines whether the checking and detection of a specific type of Channel Error is required. In other cases, it must be reset and set on a rotating schedule in certain Channel Words in order to indicate a specific type of error when it occurs, and not to indicate it when normal causes change the conditions by which the error is detected. Its use mainly depends on the type of Subsets with which the channel is functioning. The use of this bit in connection with different types of Channel Errors is explained in Section XII-1. The program must set and reset this bit (or initiate the change of its setting when the Action Delay feature is used). (In the CWD of a half-duplex high-speed channel functioning with FM Subsets, this bit has a different function.)

c. SEND-RECEIVE BIT (bit 12)

Used in Receive and Send Statuses.

Same as in Start-Stop Channel Words; a One indicates Send Status and a Zero, Receive Status. The program must set this bit (or initiate the change of its setting when the Status Change feature is used.)

INACTIVE STATUS FOR HIGH-SPEED CHANNELS

A high-speed channel always transmits Synchronous characters. It may be inactivated either through its regular Channel Word or its Error Channel Word, depending on which one is accessed and read out of Control Storage into the Control Storage Data Register under the existing conditions. The program has constant control over the status of every channel and can inactivate a high-speed channel by placing the regular Channel Word or the Error Channel Word in Hold Status at any given phase of the operation. In a regular Synchronous Channel Word, both bit 26 and bit 14 must be

reset to Zeros, unless bit position 26 already has a Zero setting, originated either in the basic setting or prior to the execution of the routine which performs the inactivation. However, bit 14 must be reset to Zero every time , because it is always One in a Synchronous Channel Word in operational (i. e. active) status.

From another logical approach, every CWD in Hold Status is a Start-Stop CWD, because the change of bit 14 in a Synchronous CWD from One to Zero creates a Start-Stop CWD for the machine logic. Therefore, it can be stated that the machine logic inactivates a channel only if the Start-Stop CWD associated with the channel contains a logical Zero in the Not Hold bit position.

In an Error Channel Word, only bit position 26 must be reset to Zero, because an ECWD is a Start-Stop type word; therefore, bit 14 is always a Zero.

C. THE SETTING OF THE CHANNEL WORD

The CWD must be set by the program. Normally, an image CWD is set and loaded into Process Storage with the program for each variation the specific system application may call for. Whenever a new CWD format is necessary, the program accesses the proper preset image CWD, inserts an Address in the Address Field, and moves it into the predetermined Control Storage CWD location assigned to the specific channel. However, the Address could also be inserted in the Address Field after the image had been stored in Control Storage and become the new CWD. In any case, the Address must be inserted before the new CWD begins its functions. The Channel Word formats are shown in the Appendices as follows:

Appendix I, Chart 4 - General CWD format
Appendix III, Chart 1 - Start-Stop Receive CWD format
Appendix III, Chart 2 - Start-Stop Send CWD format
Appendix III, Chart 3 - Synchronous Receive CWD format
Appendix III, Chart 4 - Synchronous Send CWD format

The preceding Chapter (3, B) contains all information necessary for the programmer to plan the basic settings and the desired changes of his Channel Words. It is recommended that the programmer set each unused bit in every Channel Word to Zero in order to avoid any possible misinterpretation by the machine logic and to express consistency in the program.

It may be desirable to take care of certain changes in Channel Words by the Action Delay feature. Since only one prepared section (i. e. "Y" field)

may be kept in Process Storage for all Channel Words assigned to channels scanned through the same Process Control Scanning Point, the programmer must be very careful in deciding when and how to use the feature. On the other hand, programs based on carefully analyzed operational specifications may utilize a Y field for making several types of changes in the same Channel Word if the change in the Y field itself, between each two consecutive applications can possibly be programmed with absolute safety. In addition to the Action Delay Character, other special characters, namely, the Status Change and Sending Delay Characters, and the different Adapter Synchronization bit patterns are also available for the programmer to initiate the desired functional status changes in the Channel Words.

D. ADAPTER SYNCHRONIZATION

Adapter Synchronization, meaning to synchronize the Channel Adapter with the Process Control, is a procedure consisting of a series of automatic functions necessary to assure correct data transmission. It must be initiated by the program each time the status of a channel (Channel Word) is about to change from Send to Receive, or from Receive to Send; or before a new transmission begins on a channel which has previously been inactive, but its Send or Receive status is unchanged. The Adapter Synchronization procedure results in the adequate settings of certain hardware triggers, Adapter Control Interface lines, and the setting of the four low-order bits in the Character Control field of the CWD to Zeros, making it available to be used in the following data transmission as a Bit Counter, if necessary. If characters will be sent (Send Status) on the channel, one of the functions of Adapter Synchronization is to load the first output character in the Assembly Area of the CWD.

If a Status Change Character, as described in Section IV-4, is inserted in an outgoing message, the hardware automatically sets the correct Adapter Synchronization into the Character Control Field.

Adapter Synchronization should not be confused with Character Synchronization, which is described in Section IV-4, C. However, it is a rule that Character Synchronization must be established through Adapter Synchronization and cannot be set up independently.

Adapter Synchronization is initiated by setting the Character Control field (bits 19-15) of the CWD to a predetermined bit pattern. There is only one such pattern for Start-Stop Channel Words and Synchronous Send Channel Words. Several patterns are available for Synchronous Receive Channel Words, depending on whether or not Character Synchronization is to be established and, if so, which of the available methods has been chosen. According to the selected method, a certain group of functions will be generated. The various groups of generated functional statuses are called "Sequences".

The following list contains the possible cases for Adapter (and Character) Synchronization and the bit pattern to be used for each.

Description of CWD	Bit Pattern in Octal
1. Start-Stop	
Send Status	14_8
Receive Status	14_8
2. Synchronous	
Send Status	14_8
Receive Status	
a.) Sequence 1 — Adapter Synchronization going to Synchronous Receive	14_8
b.) Sequence 2 — Adapter Synchronization going to Continuous Hunt	34_8
c.) Sequence 3 — Adapter Synchronization going to Hunt and then Receive	16_8

Note: After the Adapter Synchronization has been completed, the bit pattern 17_8 places the CWD in Hunt Status, but the programmer is not concerned with this pattern because the content of the Character Control field is automatically incremented to 17_8 as the last function of the Adapter Synchronization.

d.) Sequence 4 — Adapter Synchronization going to Hunt and then Continuous Hunt. (See note for Hunt under Sequence 3.) 36_8

Sequences 3 and 4 provide a means for operating at higher efficiency than Sequences 1 and 2.

4. THE OPERATION OF CHANNEL WORDS

A. GENERAL DESCRIPTION

The operation of Channel Words may be summarized in the following four main functions:

1. Assembling and distributing characters.
2. Establishing and maintaining character synchronization, if required.
3. Controlling (i. e. establishing, maintaining, modifying and changing) the operational status of the channels.
4. Establishing and reinstating synchronization between certain hardware components.

It is difficult to draw definite lines between the functions of the various hardware components of a system. Communication and Process Control Channels, Adapters, Adapter Control Interfaces, Triggers, Hardware Registers, and Clock Devices function together in perfect coordination and synchronization. The Channel Word links a number of such components. It provides temporary buffer storage for the constantly flowing input-output data of a "TELE-PROCESSING" System and predetermined positions for control information. When in operation, it must be in the Control Storage Data Register. It is accessible for the Communication Channel through the Adapter and its Control Interface. It may also be accessed by the IBM 7750 stored program through instructions, in which case, the CWD is also read out into the Control Storage Data Register and operated on by the machine logic according to the interpretation of the decoded instruction.

The channel-status controlling and hardware synchronizing functions have already been described in Section IV - 3 (The Preparation of Channel Words). In the following chapters, the methods of character recognition, the characteristics of the different CWD statuses, the concept and types of character synchronization, and the initiations of Sending Delay, Status Change and Action Delay are presented.

B. CHARACTER RECOGNITION

The condition that a complete character has been sent from, or assembled in, the Assembly Area of the CWD must be recognized by the machine logic in order to initiate a one-cycle Character Interrupt and obtain the next output character from, or store the assembled input character in, Process Storage. These conditions are as follows:

1. Start-Stop Receive Status

Incoming bits are gated into the bit position of the Assembly Area

which is specified by the value of the Character Length. It is called the "Receiving bit position". (Example: If Character Length is 7, each bit is gated into bit position 7 of the Assembly Area.) As each bit is received, it is shifted towards the low-order bit position (bit 1); as soon as a logical One (Start bit) is sensed shifting into the low-order bit position and a logical Zero (Stop bit) shifting into the Receiving bit position (bit 7 in the example), the character is recognized and Character Interrupt is initiated. Before the character is stored, the Assembly Area is shifted again. During this last shift, the Start bit is shifted out of the Assembly Area; therefore, it will not be stored in Process Storage. The Assembly Area is cleared after the character has been stored. See Appendix IV, Chart 1 for an example of a Start-Stop type character assembly.

2. Start-Stop Send Status

The four low-order bits of the Character Control field form a binary counter called the Bit Counter. Its value is reset to Zero; the first time as the last step of the Adapter Synchronization and from then on each time a new character is obtained from Process Storage. When the channel demands a bit, the Assembly Area is shifted one position towards the low-order bit position, and the bit in the low-order bit position is "shifted out". Each time a bit is sent in this way to the Channel Adapter from the low-order bit position of the Assembly Area, the Bit Counter is incremented by one. Before the increment, the Bit Count is compared with the Character Length and if they are going to become equal after the increment, Character Interrupt is initiated. The next output character is loaded into the Assembly Area and the procedure starts from the beginning. When a character is loaded into the Assembly Area, the entire contents of the character field (i.e. all 11 bits) addressed by the Address in the CWD is moved from Process Storage into the Assembly Area.

3. Synchronous Receive Status

Incoming bits are gated into the Receiving bit position of the Assembly Area. As each bit is received, the Assembly Area is shifted towards the low-order bit position and the contents of the Bit Counter are incremented by one. Before the increment, the Bit Count is compared with the Character Length, and if they are going to be equal after the increment, Character Interrupt occurs. The assembled character is stored in Process Storage and the Bit Counter is reset to Zero. The Assembly Area is not shifted again as in Start-Stop since all bits are significant (data) bits.

4. Synchronous Send Status

Same as Start-Stop Send Status.

C. CHARACTER SYNCHRONIZATION

Character Synchronization is a method of establishing and maintaining a condition in which the bits assembled in the Assembly Area and recognized as a character are really bits "belonging" to an input character (i.e., they form one of the predetermined bit combinations of the complete series of valid characters of which a specific "code" consists).

During output operations, when characters are sent out of the IBM 7750, such a problem does not exist. The data stored in the output areas of Process Storage is checked carefully by the program; it is obvious that during Character Interrupt cycles complete characters are obtained and loaded into the Channel Words. The bits, when sent to the Channel Adapter, are counted and the next Character Interrupt is initiated when the bit count in the Bit Counter equals the Character Length. Finally, each bit transmitted on a Channel from the Channel Adapter is checked against the bit value which was saved when it was shifted out of the Assembly Area.

The input phase of the operation is quite different. Bits are transmitted to the IBM 7750 on Communication Channels which are exposed to uncontrollable outside influences. This may result in unexpected changes, such as, for example, when noise-burst-produced additional "bits" are mixed with Terminal-generated bits. The malfunction of one of the components of a network (e.g. a Subset) may produce similar undesirable results, such as loss of bits. There is no complete protection against such environmental interferences, but the undesirable results can be reduced to a minimum level by establishing and maintaining Character Synchronization. A long message may serve its purpose just as well if one or two rather insignificant characters are lost due to such environmental disturbances, or a format-controlled message may be recovered in its entirety very quickly by the retransmission of a small segment of the message if only a few characters have been destroyed. The aim is, therefore, to provide control over the possible occurrence of such environmental errors and, by some method, reduce to a minimum the possibility of losing characters or recognizing and storing false "characters", by reinstating the normal operation (i.e., the reception of complete, valid characters) as quickly as possible. It is up to the program to detect and correct the unwanted results of the environmental errors reflected in the received data.

START-STOP CHARACTERS

The received character is recognized by the Start bit shifting into low-order

bit position and the Stop bit shifting into the Receiving bit position of the Assembly Area. When an "out-of-synch" condition arises resulting from having extra (e. g. noise-burst-generated) bits in a message, the possibility of the machine accepting a considerable number of consecutive false "characters" is low. This is so because, by coincidence, each incorrectly recognized "character" must contain a logical One in the low-order, and a logical Zero in the Receiving, bit position of the Assembly Area. Most likely, most of the unwanted, and usually identical extra bits will be shifted out of the Assembly Area; very few false "characters", if any , will be recognized and stored, or only a few significant bits will be lost this way. If some bits are lost during transmission, again only one or a few false "characters" will be recognized, or a few significant bits will be shifted out of the Assembly Area and lostbecause the machine is constantly seeking the legitimate condition and will find it quickly. Therefore, it is quite obvious that either after the loss of a few bits (or even one or two characters), or after the recognition and storage of one or two false "characters" (or after a possible combination of these two unsatisfactory results), the machine will find the legitimate condition and re-establish the character synchronization very quickly. Recently obtained figures show a probability of approximately 36% for re-establishing Character Synchronization within one character time.

Consequently, for Start-Stop characters, no special method is necessary to establish synchronization because it is established and reinstated by the Start and Stop bits. This type of character may be called a "self-synchronizing" character.

SYNCHRONOUS CHARACTERS

Synchronous characters have no Start and Stop bits; therefore, special "synchronizing characters" are used to establish Character Synchronization if it is required. The bit pattern for a synchronization character may be any combination of logical Ones and Zeros and may have a maximum length of 16 bits. Only one bit pattern may be used for a Process Control Channel (i. e., Scanning Point) and not more than two for an IBM 7750. Since these characters must be recognized by the machine logic, their bit patterns must be predetermined and prepared for wired-in recognition; therefore, the programmer cannot change them.

The Synchronous Receive status of a CWD does not provide Character Synchronization. If Character Synchronization is established at the beginning of an operation, and afterward the CWD is operating in Synchronous Receive Status, the received bits are counted and each group of the required number of bits are recognized and stored in Process Storage as a character - regardless of whether they are valid or false characters.

Therefore, in some applications, it is desirable to transmit synchronization characters at predetermined intervals to secure the maintenance of Character Synchronization.

Two types of Character Synchronization are available: "HUNT" and "CONTINUOUS HUNT". Either one, or the combination of both, may be used in a system. The technical and programming personnel should decide which method is best suitable for a particular application.

Synchronous Receive Channel Words may be operated in three different statuses. One of them, the Hunt status, is an intermediate status and can be used only for establishing Character Synchronization. There are two statuses for data input operations: Synchronous Receive and Continuous Hunt. The bit patterns used for Adapter Synchronization determine the data-operational status of a CWD, as described in Section IV-3, D. The three statuses, the Synchronous Receive CWD may be operated in are described in the following paragraphs.

1. SYNCHRONOUS RECEIVE Status

   Each bit coming in is shifted toward the low-order bit position of the Assembly Area and counted in the Bit Counter. When a complete character has been received, Character Interrupt occurs. The character field in Process Storage addressed by the Address in the CWD is cleared and the character is stored. The Bit Counter is reset to Zero and the procedure starts from the beginning. Character Synchronization is not provided by this method. If required, it can be established by selecting Sequence 3 for Adapter Synchronization, but once established, it is no longer maintained. Sequence 1 provides a means for establishing Synchronous Receive status directly from Adapter Synchronization, when Character Synchronization for the following incoming transmission is not required at all.

2. HUNT Status

   In Hunt status, a 15 bit end-around shift register is formed using the 11 bit positions(bits 11-1) of the Assembly Area and the 4 bit positions (bits 26-23) of the Extended Assembly Area. In addition, the bit in the Data-In line of the Adapter Control Interface (which is the sixteenth or high-order bit) is included in the 16 bit area examined by the machine logic when searching for a synchronization character. Each bit is entered into the Receiving bit position of the Assembly Area, as specified by the Character Length and shifted

towards the low-order bit position. The low-order position of the 16 bit Assembly Area is equal to the bit position specified by the Character Length plus one. For example, if Character Length equals 7, the low-order position is bit position 8 of the CWD (i.e. CSDR). A maximum of 15 bits are maintained in this shift register. In Hunt Status, the only function of the machine logic is to search for the synchronization character; therefore, bits are not counted and no character is stored in Process Storage. As each new bit enters the Receiving bit position, the bits are shifted towards the low-order bit position as shown in FIGURE 2. The machine recognition logic examines the contents of the 16 bit Assembly Area at each new bit time. When the logic indicates that the proper synchronization pattern has been received, Character Synchronization is established, and the Hunt Status is terminated. If Sequence 3 had been initiated, the four low-order bit positions of the Character Control field are reset to Zero to function as a Bit Counter and, from this point on, the CWD is in Synchronous Receive status. However, if Sequence 4 had been initiated, the CWD is placed into Continuous Hunt status at this point, as is described under "Continuous Hunt status" below.

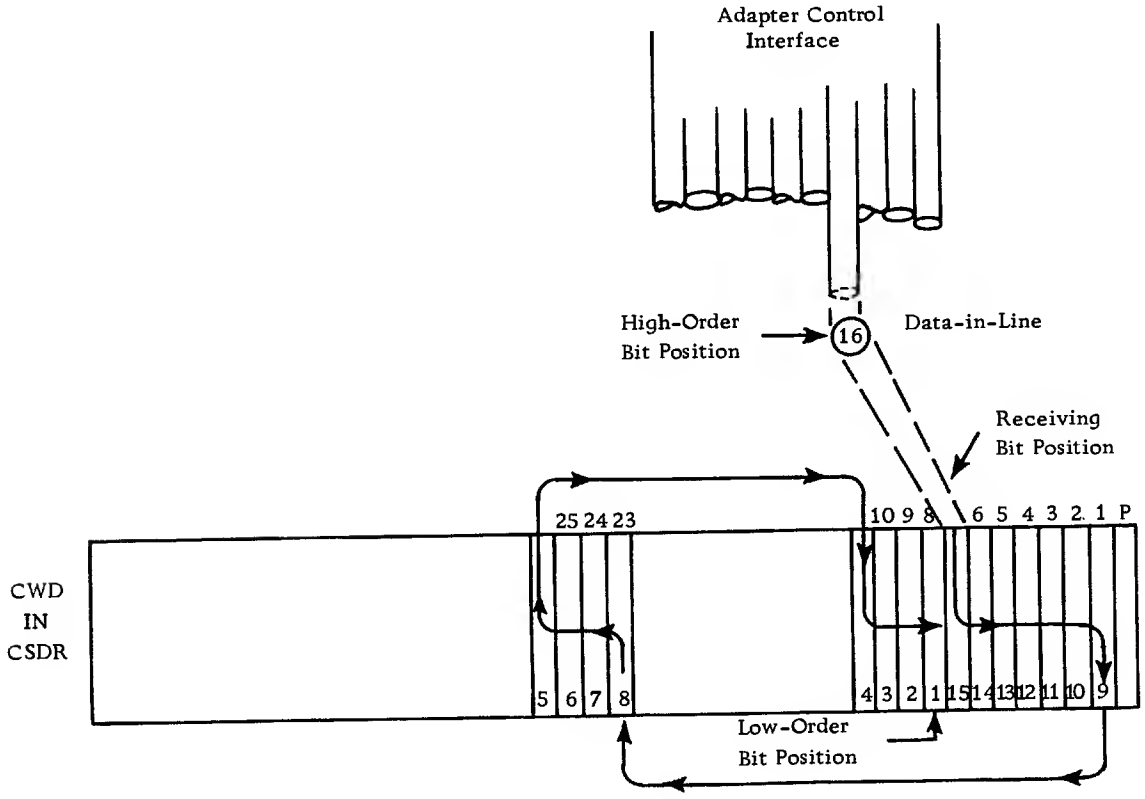


FIGURE 2: 16 Bit Assembly Area in Hunt and Continuous Hunt Status

3. CONTINUOUS HUNT Status

In Continuous Hunt status, a 16 bit Assembly Area is formed and used according to the same principles introduced under "Hunt Status" above. However, the operational procedure is different. As each bit enters the Receiving bit position of the Assembly Area, the bit count is incremented.

If a complete character has been received, Character Interrupt is initiated. The addressed character field in Process Storage is cleared and the number of bits corresponding to a character are stored before the shift is effected. However, 15 bits are maintained at all times in the 15 bit shift register. The machine recognition logic continuously examines the contents of the 16 bit Assembly Area at each new bit time. Character Synchronization is re-establishe each time the logic indicates that the synchronization character has been received. When this recognition happens, the four low-order bits of the Character Control field (i.e., Bit Counter) are reset to Zero and the count begins with the next received bit. The high-order bit (bit 19) of the Character Control field, which is interpreted as the "Continuous Hunt Bit" is never turned off; therefore, it will continue to command the machine logic to constantly check the 16 bit Assembly Area for the synchronization character.

In Continuous Hunt status, the bits of the synchronization character are stored in Process Storage as characters. The number of bits which will be stored depends on the length of the synchronization character and the Character Length. Example: a 16 bit synchronization character may be stored as two 8 bit characters if the Character Length is 8. Therefore, the programmer must make provision in the program for checking and deleting such input.

D. AUTOMATIC STATUS CHANGE FROM SEND TO RECEIVE

This feature may be used for either Start-Stop or Synchronous character-transmitting channels, but in a different way.

The change of the status of the CWD from Send to Receive may automatically be performed by the machine logic through the recognition of a special character, the STATUS CHANGE CHARACTER (SCC). When Start-Stop characters are transmitted, a second special character, the DELAY COUNT CHARACTER (DCC)must follow the SCC. When Synchronous characters are transmitted, no DCC exists. In each case, the status change is initiated by the program through the insertion of the Status Change Character (and the DCC when Start-Stop characters are transmitted) at the

end of an outgoing message or polling message. This feature may also be used as an independent procedure to control maximum response time for a Terminal polled on the send line of a full-duplex channel in Start-Stop character operation. The CWD associated with the receive line of the channel is placed in "Send Status" to cause these two characters (SCC and DCC) to be moved one by one into the CWD's Assembly Area. The SCC returns the line to its normal operating status of receive, and the DCC represents the time within which the Terminal must respond.

The Status Change feature may be used on half-duplex channels when polling a Terminal, normally right after the completion of an outgoing message, to condition the Channel Adapter (the Adapter Control Interface) and the CWD for input operation. If the Terminal is prepared for transmission, it will start transmitting; if not, it will send a predetermined character or a short message, indicating that it is not prepared for transmission (i.e., it has nothing to transmit at the time it is polled). In some cases, on high-speed channels, an "acknowledgement" signal or character may be required from the polled Terminal, regardless of the following transmission. The program must make provisions for initiating investigation and proper action (i.e., printed or typed messages for operating personnel) for cases when neither of these responses will occur within the allowed time period after the first, or sometimes repeated, polling. The feature may also be used simply for terminating message sending when, for example, the available buffer space is low in the IBM 7750; therefore, no input data is called for and no more output data is prepared in the output area for any terminal on that channel.

Similarly, the feature may be used on the outgoing line of a full-duplex channel, for terminating message sending or after polling when no immediate output operation is due. It may be used on the incoming line of a full-duplex channel in Start-Stop type character operation as described earlier in this Chapter.

The bit pattern of the Status Change Character is: 3776_8 . In Start-Stop character operation, it must be followed by a DCC. For Synchronous character operation, there is no DCC. During the Character Interrupt cycle when the SCC is obtained from Process Storage, the machine logic recognizes it in the Assembly Area as an SCC. However, the following generated functions are different, if the recognition occurs in the Assembly Area of a Start-Stop Send or of a Synchronous Send CWD:

> In a Start-Stop Send CWD, the Delay Bit (bit 25) is automatically turned on upon the recognition of an SCC. The next time the same CWD is accessed, another Character Interrupt is initiated, the Delay Count Character is loaded into the Assembly Area, the

Character Control field is set to 14_8 (i.e., for Adapter Synchronization), and the low-order bit (bit 12) of the Status Macro (Send-Receive Bit) is set to a logical Zero (i.e. to Receive). From this point on, each time this CWD is accessed and a new bit time occurs, the binary number in the Assembly Area (i.e., the Delay Count Character) is decremented by one. When it is decremented to all Zeros, the high-order bit (bit 26) of the Status Micro (i.e., the Not Hold Bit) is reset to a logical Zero, placing the CWD in Hold Status, and, by this, the channel into inactive status. If the first bit (i.e., the Start bit) of a character is received on the channel before the Delay Count Character is decremented to all Zeros, the delay is terminated, the Delay Bit is turned off, the Assembly Area is cleared, the bit is entered into the Receiving bit position of the Assembly Area, and the character assembly begins.

The Delay Count must be figured by the programmer in bit times. Example: if the transmission speed (i.e., bit rate) on the channel is 75 bps and a two-second delay is desired, the Delay Count Character must be $150_{10} = 226_8$. The maximum delay that may be initiated is $2045_{10} = 3775_8$ bit times.

If it is intended to put the Status Change into effect without time delay, an all Zero Delay Count Character must be used; it cannot be omitted.

Neither the SCC nor the DCC can be located in the Block Control Character (i.e., the last character position) of a block.

In a Synchronous Send CWD, upon the recognition of the SCC, the four low-order bits of the Character Control field are automatically set to 16_8 (i.e., Adapter Synchronization) and the low-order bit of the Status Macro (Send Receive Bit) to a logical Zero (i.e., to Receive). If the high-order bit of the Character Control Field (i.e., the Continuous Hunt Bit) is a logical Zero, then Sequence 3 is initiated. If it is a logical One, then Sequence 4 is initiated.

E. SENDING DELAY

This feature may be used in Start-Stop character transmitting output operations only.

During output operations, certain types of Terminal sets require periodical time allowances for positioning their components. Example: a typewriter-type set needs time for carriage return and line positioning. Therefore, besides the functional characters controlling these movements, other

characters become necessary to cause a delay in transmitting and thereby eliminate the otherwise resulting character losses. Such special characters are the SENDING DELAY CHARACTER (SDC) and the following Delay Count Character (DCC).

Whenever such delay is necessary, the program must insert these characters in the message when it is stored in the output area. The necessity for the insertion of such characters may be entirely controlled by the IBM 7750 Normal Mode Program through character count while the data is prepared for the output area, or may be indicated in the computer-processed data by certain predetermined characters. In the latter case, the IBM 7750 Normal Mode Program must identify these "indicators" and insert the proper equivalents in the outgoing message while preparing output data from the data received from the Processor.

The bit pattern of the Sending Delay Character is 3777_8. During the Character Interrupt cycle when this character is obtained from Process Storage, the machine logic recognizes it in the Assembly Area as an SDC and turns the Delay Bit (bit 25) on. While the Delay Bit is "on", no bits are shifted out of the Assembly Area (i. e., no bits are sent to the channel). The next time the same CWD is accessed, another Character Interrupt is initiated, and the Delay Count Character is loaded into the Assembly Area. From here on, each time this CWD is accessed and a new bit time occurs, the Delay Count Character is decremented by one. When the DCC is decremented to all Zeros, the Delay Bit is automatically turned off; the next time the same CWD is accessed again,the next output character is obtained from Process Storage and the normal outgoing transmission procedure continues.

The rules for the delay time configuration and for the maximum delay are the same as for the Delay Count Character following an SCC. Neither the SDC nor the DCC can be located in a Block Control Character.

For Synchronous character operation, the Sending Delay feature is not available. If a "delay" is desired, for any reason, dummy (blank) characters can be inserted in the output data, but they will be transmitted out on the channel.

F. ACTION DELAY

This feature may be used in all types of Input-Output operations.

Sometimes it is desired to make a change in the operational and/or functional status of a Channel Word sooner than the program would be able to do it. Such "action" can be achieved by the application of

the Action Delay feature. It is rather a "speed-up service" than a delay; its name has been derived from the number of character times that have to elapse, if any, before the "action" is performed.

The operational and functional status of a Communication Channel is controlled by the Control Area in its associated Channel Word. When this Area, or certain fields in this Area, are changed, the status is changed. A change in the setting of the Send-Receive Bit in the Status Macro, accompanied by the loading of the proper Adapter Synchronization bit pattern into the Character Control field, reverses the operational status of a channel. In some cases, a quick change is required in the setting of the Control Bit in connection with error detection. Such a quick change may even be required on a reset-set alternating schedule. In order to prepare new contents for certain fields of a Channel Word, an eleven bit character field is reserved in Process Storage, with a predetermined address, and assigned to each of the Process Control Scanning Points. These character fields are called Y fields. The contents of the proper Y field are loaded in a Channel Word when conditions are met and replace the previous contents of bit positions 22-12. The bit positions in a Y field correspond to the Sequence Counter, Action Delay Bit, Character Control, and Status Macro fields of the Channel Word. They may be prepared by the program in any desired manner. As soon as the contents of this field are loaded in a Channel Word, the operational or functional status, or both, will be changed.

There are a maximum of 16 Y fields in Process Storage because the maximum number of Scanning Points on the Process Control are 16. The Y fields are located in the 16 high address character fields in upper Process Storage. When accessed, they are addressed by the position of the Process Control Channel Scanner. (i.e., only the four low-order bits must be addressed because the 12 high-order bits of the 16 bit address are identical for all 16 Y fields and are permanently wired in.) Since one Y field is assigned to each Process Control Scanning Point, all Channel Words assigned to Communication Channels which are scanned through the same Process Control Scanning Point may be affected by the same Y field.

In the Channel Word, the Action Delay Bit (bit 20) and the Sequence Counter (bits 22-21) serve the purpose of demanding the Action Delay feature by the program and setting a time limit, expressed in character-times, for its execution. The recognition of a special character, the Action Delay Character (ADC) in the Assembly Area of the Channel Word, starts the procedure.

The program, in order to utilize this feature, must turn the Action Delay

Bit on (i. e., set it to One) and set the Sequence Counter to a binary number equal to three minus the desired number of character-delay times. The highest number of character-delay times is obviously three. Since 3 minus 3 equals 0, an all Zero setting in the Sequence Counter is necessary if the contents of the Y field are required to be loaded in the CWD after three character times have elapsed from the recognition of the ADC. While the Action Delay Bit is "on", the machine logic searches for the Action Delay Character in the Assembly Area. As soon as the ADC is recognized, the Action Delay Bit is turned off and the Sequence Counter is incremented by one. Each succeeding Character Interrupt cycle will advance the Sequence Counter. When the contents of the Sequence Counter become Zero, which happens when 11_2 is incremented by one to 100_2 with the drop of the carry, the associated Y field is accessed and loaded into the Channel Word. If immediate action is required, without any character-time delay, 3 minus 0 equals 3 (i. e., 11_2) must be placed in the Sequence Counter.

The Action Delay feature may be equally used for Start-Stop and for Synchronous channels in both Receive and Send Status. Consequently, the ADC may be transmitted by a Terminal or may be obtained from the Process Storage output area of a channel.

The Action Delay Character must have a predetermined bit configuration because it must be recognized through wired-in logic. The number of bits in the ADC generally must correspond to the number of bits in a Data Character. However, if a system transmits characters of different bit lengths, there may be a possibility, in output operations only, that the same ADC may be used for all Channel Words. This may be so because all eleven bits of an output character field are loaded into the Assembly Area regardless of the actual data character length, and the machine logic, each time a new character is loaded, examines all eleven bits of the Assembly Area when searching for the ADC. Therefore, it is forced to recognize the ADC. Permission to use this feature in such an unusual way must be obtained from technical personnel, because certain type of channels make this impossible.

By storing a One in bit position 9 of a Y field and the calculated number, resulting in the desired character-time delay, in bit positions 11-10 (corresponding to bit positions 20 and 22-21 in the Channel Word respectively) the feature may be automatically demanded (repeatedly) each time the Y field is loaded into the Channel Word.

There is only one ADC available for an IBM 7750.

(The name "Y field" is originated from the association of bit positions 22-12 in the CSDR with the Y register, located in Process Words. When a PWD is read into the CSDR, bit positions 22-12 correspond to register Y.).

5. THE INITIATION AND TERMINATION OF TRANSMISSION

Central "TELE-PROCESSING" Installations not working on a 24 hour schedule must be initialized for correct data transmission each time the new operation period begins. Similarly, when the operation is disconnected for maintenance purposes, or as a result of an unexpected failure, adequate preparation is necessary to assure efficient operation when it is reinstated. This initialization includes the Adapter Synchronization for each communication channel and the Character Synchronization, initiated through the Adapter Synchronization whenever necessary, as well as the housekeeping requirements mentioned below. The proper routine for such "system initialization" must be available for execution at any time.

The rest of the preparation which should be taken care of in the initialization routines, such as providing the Receive Channel Words with input area addresses for their first incoming characters, storing Addresses and Limits (when required as part of the initialization) into the uniquely assigned area-controlling Limit Words and Limit-Word-Chain-controlling Master Limit Words, and the numerous other necessary initialization steps will certainly appear rather familiar to programmers. These preparations are based on the same principles they followed in writing computer programs before becoming acquainted with the IBM 7750.

The initiation and termination of message transmission during normal operation is quite simple. The functions the programmer is mainly concerned with as initiative steps for oncoming transmissions are Adapter Synchronization with or without "built in" Character Synchronization, Address loading into the Channel Words when necessary, turning Not Hold Bits on and Delay Bits off after a Delay Count expiration and, in addition, setting the Send-Receive bits from Receive to Send after "outgoing only" channels are placed temporarily into inactive status by the Status Change feature.

The termination of input message transmissions is indicated by unique "End of Transmission" or "End of Message" characters. These are transmitted by Terminals and can only be detected through the program. When Sending or Polling, the application of Status Change Characters (with Delay Count Characters set to the required value in bit times or to all Zeros, or without Delay Count Characters, as desired within the permitted limits) provides the means for terminating transmission. Whenever the output area of a channel has been temporarily exhausted and it is not a half-duplex channel, or even if

it is a half-duplex channel but no incoming data transmission is desired (i.e., no Terminal is polled on the channel), the transmission must be terminated. (On low-speed channels, the termination is associated with the inactivation of the channel.) If no provision is made for termination, the IBM 7750 keeps sending out the contents of the remaining consecutive character positions in the last block of the output area and will request Channel Service Mode at the end of the block. This would result in an "endless" output operation.

The termination of outgoing transmission is different for Start-Stop and Synchronous character-transmitting channels. A Start-Stop character-transmitting channel can be placed into Receive Status and inactivated by the application of an SCC and all Zero DCC. A Synchronous character-transmitting channel may be placed into Hunt or Continuous Hunt (Receive) status by the use of an SCC depending on the existing setting of the Continuous Hunt Bit of the CWD (logical Zero for Hunt, logical One for Continuous Hunt), and may be left in this status, since no incoming character is expected on the channel. However, if desired, a Synchronous character-transmitting channel can be inactivated, following the application of the SCC by the programmed resetting of the high-order bit position of the Status Macro (bit 14) to a logical Zero. As previously explained, in order to inactivate a channel, both bits 26 and 14 must be reset to logical Zeros in the CWD. In the present case, bit 26 is reset to Zero when the CWD is placed into Hunt or Continuous Hunt status because the Extended Assembly Area (bits 26-23) is automatically cleared; therefore, the programmer is not concerned with its resetting; only the resetting of bit 14 must be programmed.

SECTION V

LOGICAL ORGANIZATION OF THE IBM 7750 STORED PROGRAM

The general organization principles of the stored program were introduced in Section II. The present section examines, from a logical approach, what functions of the stored program complex should be taken care of in each of the five programmed modes.

As already mentioned, the programming personnel are not completely tied to binding rules when organizing the program. However, there are several functions that must be taken care of in specific Mode Programs, either because the machine logic will demand the execution of certain Mode Programs when specific conditions are recognized, or because the correct logical approach almost imperatively requires that certain conditions be satisfied in a specific phase of the operation. The available priority processing levels offer an easy and efficient way to do so by giving preferences to the execution of such routines for greater program efficiency and higher hardware utilization. The necessary and logically chosen routines to be programmed in each mode are given in the following chapters.

1. THE SERVICE MODE PROGRAM

There are three ways to initiate the execution of this program. It can be requested automatically through the machine logic upon the detection of Communication Channel Errors; similarly, the request is automatic following a Process Control Error detection if the Check Switch on the Operator's Panel is set to Service Mode instead of Stop (most likely, this will be the "standard setting" of this switch in most applications.) Service Mode may also be initiated manually from the Operator's Panel. This method most likely will be used when checking out the machine during maintenance time. There seems to be little reason to use the third method, which is programmed request. The corrective and recovery routines for common data errors caused by the originating terminals, their operators, or by out of synch or other transmission failures should be located in the Normal Mode Program. The detection of such errors will usually happen in the Normal Mode anyway; therefore, no justifiable reason can be given to switch the operation into the highest priority mode. Errors detected in the data transmission between the IBM 7750 and the Processor should be corrected in the Out or In Mode, whichever preceded the erroneous transmission-producing Copy Mode. Consequently, the Service Mode Program should be more or less restricted to the identification of Communication Channel and Process Control Errors; but, even the corrective routines for these errors should be placed in the Normal Mode Program to avoid keeping the machine unnecessarily in the highest priority mode, and thereby delaying

the execution of other requested priority mode programs. However, if desired, the programmer may build his own "priority error routine" for such cases into the Normal Mode Program by saving the contents of the Normal Mode Process Word (i. e., the contents of the IC and X, Y, Z registers) and placing the location address of the first instruction of the selected corrective routine (still in the Service Mode Program) in the Instruction Counter immediately after the identification of the specific error. At the end of the selected error routine, the Normal Mode Process Word must be so restored that the execution of the interrupted routine will nowcontinue. The Service Mode operation must always be terminated by the Service Mode Program. Chapters 1E and 2D in Section XII contain more information concerning the programming of the Service Mode.

2. THE CHANNEL SERVICE MODE PROGRAM

This mode is requested automatically each time an "End Of Block" condition is recognized by sensing five low-order One bits in a 16 bit character address in the Address field of a Channel Word. Consequently, at least two routines must be programmed in this mode: the assignment of a new empty block for input and the next data-containing block for output operation. Beyond these two routines, the programmer may find it desirable to place some other routines in this Mode Program to take advantage of the automatic "End Of Block" detecting and Channel Service Mode requesting features of the IBM 7750. For example, when polling is desirable on a full-duplex channel before the current outgoing data transmission is completed, such an application would be to store the characters of a polling message in the last data character positions of a block and to insert this block segment between two output-data blocks. However, careful scaling by the programmer is required as to how far he may go in his specific system application in "loading" the Channel Service Mode with extra routines without crossing the "danger line"and risking the loss of incoming characters by not keeping this Mode constantly available to fulfill its original purpose. This mode may also be requested by programming, but there is no reason for doing so since its operating function is as described above.

The termination of the Channel Service Mode operation must be performed by the Channel Service Mode Program.

3. THE OUT MODE PROGRAM

Control may be turned over to this mode by the IBM 7750 program or by the Processor program through special Commands sent through the Corporate Simplex I/O Interface. The IBM 7750 program may request this mode only for the execution of routines placed in this Mode Program for convenience, but data can be transferred to the Processor in a following Copy Mode operation only if, previous to the request for Copy Mode, the Out Mode Bit in the Mode Request Register has been turned on by the machine logic upon

sensing a Command received from the Processor. (See Section VI for details on data transfer between the IBM 7750 and the Processor.) The preparation of the Copy Word for the following data transfer to the Processor, the clearing of the Interface Data Register, and the request for the Copy Mode are the functions which the program of this mode must take care of before the Copy Mode operation. After the data transfer, when the Copy Mode operation has automatically been terminated, proper checking must be done in the Out Mode to determine whether the transmission was a complete, valid one. If any error in Copy Mode, such as an incomplete transmission, is detected, the adequate correction must be initiated. If checking shows the completion of a satisfactory transmission, the contents of the Interface Control Register must be reset to all Zeros and the Out Mode request bit must be turned off.

The programmer must decide how much preparation for the Copy Mode data transfer will be included within the Out Mode Program and will be performed prior to the actual data transfer. He may take care of the entire preparation in this Mode by setting up his Copy Word in a Process Storage location, then moving the contents of this word into the predetermined Control Storage location of the Copy Word; or he may simply move to Control Storage the contents of the next image Copy Word which has already been set up in the chain of such words controlled by a Limit Word while processing data in the Normal Mode. The image Copy Word, which is a Limit Word, must have in its Address field, the address of the first character of the message to be sent to the Processor. The Limit field must contain the address plus one of the last character of the outgoing message.* The preparation also includes setting the Z Register in the Copy Word. Depending on the specification for the data transfer operations, the status of the Sense Bit in the Interface Control Register may well influence which of several types of preparations should be effected. In some applications, even short control messages may be transmitted between the IBM 7750 and the Processor to determine specifications for the following data transfer. When turning on the Copy Mode Bit in the Mode Request Register, the program must not turn off the Out Mode Bit because the checking of the terminated transmission must be done in the same Mode Program. Upon the termination of transmission, the Copy Mode Bit is automatically turned off and control is returned to the Out Mode. If the check shows a satisfactory completion of data-transfer and the following necessary signal exchanges with the Processor are completed, the Out Mode Bit must be turned off by the Out Mode program. If an error is detected, and again the required signal exchange with the Processor is performed, preparations should be made for a repeated or corrective transmission before turning the Out Bit off. As soon as the expected Command from the Processor turns it on again, Copy Mode will be requested to perform the corrective transmission.

If a satisfactory transmission cannot be achieved through repeated operations, an error routine must be accessed and executed, usually in the Normal Mode, after terminating the Out Mode operation.

* (The Limit must not have five one's in the low-order bits).

The Out Mode Program,once control has been returned by the automatic termination of the Copy Mode, checks for complete successful transmission by comparing the Address and Limit of the Copy Word used in transmission. If they compare, the transmission was complete. If they do not compare, the program must check the Stop Bit of the Interface Control Register to see if the Processor halted transmission. If the bit is on signifying that the Processor had halted transmission, the systems specifications will dictate which course of action to follow. If it is not on, the Program assumes that transmission was halted due to "time out in Register Z" and branches to an error routine. (This "time out in Register Z" is explained in Section VI.)

It is up to the programmer to decide what other routines he would place in the Out Mode Program, besides the necessary ones described above. Even routines logically not belonging to this complex may be implemented in the Out Mode Program in order to take advantage of its higher priority. The interrogation of certain control characters, called indicators or program switches, will direct the operation to the proper routine each time the Out Mode is entered. However, good judgment must be used to organize the Out Mode Program within the limits necessary to maintain efficient operation for both the Processor and the IBM 7750 and to avoid building up unnecessary data queues which might result in keeping data in the output area for the Processor while using the Out Mode for the execution of some neutral program routines.

4. THE IN MODE PROGRAM

The operational and organizational principles for the In Mode are the same as those for the Out Mode. Therefore, only the differences between the two modes will be listed without repeating all the details. Again the necessity of clearing the Interface Data Register must be stressed.

The Copy Word must be prepared to define an available Process Storage area for storing the characters that will be transmitted from the Processor to the IBM 7750. The storage area must be sufficiently long to accept maximum length messages.

The status of the Control Bit,instead of the Sense Bit,in the Interface Control Register must be checked for the type of preparation required according to the program specifications.

Transmission should be terminated by the Stop signal sent by the Processor. It can be checked by interrogating the status of the Stop Bit in the Interface Control Register. A termination by an equal compare between the Address and the Limit in the Copy Word normally indicates an incomplete transmission. Obviously, the Limit in the Copy Word must be set one character

location higher than the last character field in the input area; otherwise the transmission will be terminated by an equal compare before the last character can be transmitted from the Processor, if the data transmitted is a maximum size record. If the Stop Bit is "on" in the Interface Control Register and, at the same time, the Address and Limit fields in the Copy Word contain identical addresses, the transmission has been satisfactorily completed. Termination may also result from "time out in Register Z".

5. ADDITIONAL PROGRAMMING ASPECTS FOR THE OUT AND IN MODES

For both the Out and In Modes, additional bits (End, Unusual End,and End Response Bits) in the Interface Control Register serve the purpose of communicating between the IBM 7750 and the Processor in order to complete the checking of the status of a terminated or interrupted transmission.

The Attention Bit serves the purpose of signaling the Processor when the IBM 7750 program intends to initiate a transmission. It should be turned on (i. e., set to logical One) in the Normal Program and then checked at certain intervals in the Normal Program loop to see if it has been turned off by the Processor. A count, predetermined by the program, is set for the number of times the bit is to be checked. The count is decremented by one each time a check is made. This check continues until the Processor responds or the count is decremented to zero. If the decremented count reaches zero, an error routine is accessed. If the Processor responds, the Attention Bit is turned off and, depending upon the meaning assigned to the Attention signal by the program specifications, either the Out Mode or the In Mode Bit in the MRR is turned on. (At the same time the Sense Bit or the Control Bit may or may not be turned on in the IFCR respectively). The Attention Bit may not be set through a Processor signal; it may only be reset (i. e., to a logical Zero). It may be set or reset by the IBM 7750 program.

6. THE NORMAL MODE PROGRAM

This may be called the "main program" of the IBM 7750 because all processing functions not imperatively, or by logical necessity, programmed into one of the priority modes may be implemented into this program. Some exceptions may be exercised by the programmer by placing certain routines, for convenience, into priority programs as described before. However, the bulk of the programmed functions in all applications will be implemented in the Normal Mode Program.

The Normal Mode Program is never requested and never turned off because it has no bit position in either the Mode Request Register or the Mode Status Register. Whenever no mode request bit is "on" in the Mode Request Register (i. e., all 5 bit positions contain logical Zeros) and consequently the setting of the Mode Status Register is identical, the IBM 7750 operates in the Normal Mode.

The organization of the Normal Mode Program may be different for each application , depending on the functional requirements of the specific system. The programmer must scale the factors of memory occupation and execution time as pri me elements when designing the program. The closest coordination with the Processor programming personnel is essential from the very beginning. The utilization of the two units, the Processor and the IBM 7750, must be brought into a logical balance, and consideration must be given to the forecasted, or deduction based, expected expansion within a reasonable future time period. Efficiency in programming should result not only in the efficient organization and coding of both the Processor and the IBM 7750 programs, but the functions not specifically belonging to either of the machines should be reasonably divided between the two stored program computers, considering their internal organizations and processing capabilities. It would indicate poor judgment, for example, to fill a large portion of the IBM 7750 Process Storage with numerous sizeable tables and sophisticated program routines for taking care of complicated arithmetic computations, when the Processor is able to perform the same operations with little extra-time consumption and practically no extra core memory usage.

After the specifications have been completed by the systems staff, the programmers must organize the IBM 7750 program to their best judgment. However, as pointed out in Section I, the operational principles call for a unique programming method to assure the efficient servicing of all Communication Channels . It is obvious that separate programs should not be written for processing data in an identical manner in areas assigned to different Communication Channels, but the same processing functions must be taken care of by one single program routine for all input, output, or processing areas. Therefore, with the exception of a general initialization at the beginning, as required, the program must consist of numerous routines linked together by some logical control, and each routine must be available for each area that uses it. Whenever an area is accessed by the program, the status of the area determines which specific program routine must be executed. Since the program proceeds from area to area and the same routine may be executed for the same area several times, even during the processing of one single message, the Normal Mode Program must be organized on an "area scanning" basis. This means that the program must examine the input and output areas assigned to Communication Channels, through the area-controlling Limit Words, and also the input area for the Processor; it must then perform the necessary programmed functions according to the status of the accessed area. This must be done in a sequence predetermined by the programmer. The output area for the Processor is handled differently because the data to be transmitted to the computer will

be stored in different areas as the processing of the input data proceeds. Normally, Copy Word images will be set up for each information unit when it is completed; these will be located in a chain controlled by a Limit Word. Therefore, the only time the program must perform further operations on these words is when a Command from the Processor turns control to the Out Mode. When this happens, the current first word of this chain will be moved into Control Storage to become the Copy Word for the next transmission. Again, after the transmission is completed, it is checked by the IBM 7750 program, and when its validity is established through communication with the Processor, the exhausted output area for the Processor is released, and the used Copy Word image made available for a new area in the Out Mode Program. Checking of the output areas for the Processor should not be included in the area scanning schedule of the Normal Mode Program if transfer of data is under control of the Processor. In this case, the functions to be performed on these areas before and after the actual transmission in the Copy Mode will be programmed in the Out Mode Program. If data transfers may be initiated by the IBM 7750, a check of the output area for the processor must be included in the Normal Mode Program.

The so-called "process areas" are intermediate areas between the input and output areas assigned to channels. Each such area is used as storage for already processed input characters. When the area contains a complete processed message, it becomes one of the output areas for the Processor. It is placed under the control of the Limit Word controlling the output areas for the Processor by placing the first character address and the last character address plus one into the next available word in the Copy Word image chain. At this point, the process area controlling Limit Word is provided with a new Address for the next message.

Another type of "process area" is the area designated for storing the processed versions of input characters received from the Processor. This type of process area will be attached to output queues when the processing of each message is completed. All process areas are normally accessed by the Normal Mode Program through cross references; therefore, the area "scanning" schedule should be limited to the input and output areas assigned to channels through their Channel Words, and to the input area for the Processor. However, the checking of the input and output areas cannot be completely separated from each other for full-duplex and cannot be separated at all for half-duplex channels. Only simplex channels and so-called "basic control channels"(i.e., half-duplex channels transmitting data in the same direction at all times)

have either input or output areas, depending on their assignments, and may not have both. A half-duplex channel, when active, may either be in Receive Status or in Send Status, and one single Channel Word controls the alternate two way traffic; therefore, only one reference is used. For logical reasons, the input area is normally accessed in the "area scanning" schedule since this area calls for more processing activity. The output area may be checked through cross reference when the status of the input area indicates that the channel is available for outgoing transmission. The two areas of a full-duplex channel may be checked through one reference which, as stated above, should be the input area. The output area can be checked through cross reference each time the input area is checked or may be checked regularly in the "area scanning" schedule. In the latter case, the Send Channel Word must also be accessed through cross reference and checked in order to take the proper action when the Receive Line is found to be idle and polling becomes necessary.

The outgoing message in an output area is prepared by the "one shot" execution of a routine which processes, without interruption, a complete message from the input area for the Processor.

In the Normal Mode, program linkage is provided by recording in a table the address of the first instruction of the next sequential routine to be executed upon returning to the same area. (See Section XI-4). The decision,as to the sequence and the frequency with which to process each area,must be made during the program organization,and the program must be written correspondingly. Changes must be taken care of by program maintenance. Naturally, areas of high-speed Communication Channels should be "scanned" more frequently than low-speed channel areas. Areas of channels with considerably higher utilization factors (i. e., higher traffic volumes) should also be examined more often than others. A flexible way to organize the Normal Mode Program is to assign successive identification numbers in descending order to the areas to be "scanned". The Channel Words provide a good basis for the assignment. Areas desired to be serviced more often than others, such as the areas of more active channels and the input area for the Processor, may each have several numbers assigned. This number will be decremented each time when leaving an area in order to derive the identification number of the next area to be examined. The number must be reset to the original starting value each time a "scanning round" is completed.

The application of tables provides a powerful and flexible method for controlling every phase of the Normal Mode operation for all desired areas. Tables containing references and cross references, indexed by the preassigned area identification numbers and by second or even

higher level identifiers derived from the basic tables, can be the basis on which the operation can be controlled so that each channel receives the required service on a predesigned schedule. (See Section XI-4).

Since all input, output,and process areas are serviced by the same routines (i. e., one routine per function), only the Control Storage address of the Channel Word and the Process Storage location addresses of the area controlling Limit Words must be changed in the affected instructions each time the routine is to be executed for a new area. Therefore, each routine must be initialized each time it is to be executed. This consists of replacing the address of the Limit Words accessed throughout the previous execution of the routine and the address of the Channel Word (in the Storage to Storage instructions) with the current addresses.

More than a general concept of the Normal Mode Program and its logical organization cannot be given. There are no rules. The Network configurations, the programmed functions, the sequence and frequency of the area scanning, the tables used, and many other aspects will vary from application to application.

SECTION VI

TRANSFER OF DATA TO AND FROM THE PROCESSOR

Interface - Interface Control Register - Interface Data Register - Copy Mode Operation.

1. INTERFACE. In the operation of a Data Processing System, data transfer between two units is performed through the Interface by which these units are connected to each other. Therefore, the type of Interface used provides the decisive factors in determining the prerequisites which must be satisfied before data transfer can occur.

In the interest of simplicity, an Interface will be described as consisting of a number of lines (i. e., wires), each having a specific function assigned by wiring. The wired-in logics of both hardware units will recognize certain conditions and initiate the required functions performed through the proper line(s) of the Interface.

The IBM 7750 is connected to its associated computer by a Simplex I/O Interface. By using the Simplex I/O Interface, transmission of data between the IBM 7750 and the Processor can be performed only if the Processor enables the transmission lines to be used for such data transfer (i. e. establishes the hardware connections for those Interface lines, to enable them to carry the data character composing bits from one unit to another). The direction of the data transfer is also determined by the Processor. These conditions are established by the Processor through one of four possible signals, called Commands. These Commands are: Read, Sense, Write, and Control. The request bit for the Out Mode is turned on by a Read or a Sense Command, and for the In Mode by a Write or a Control Command, in the IBM 7750 Mode Request Register. In addition, the Sense Command turns on the Sense Bit, and the Control Command turns on the Control Bit in the Interface Control Register, to indicate a special condition which must be satisfied for the oncoming transmission. The program specifications must determine what programmed actions must be taken in such cases. These four Commands also result in the setting of certain hardware triggers; some of these triggers are used to establish and maintain an exchange of communications in the form of signals between the IBM 7750 and the Processor for the oncoming data transfer; others are used to constantly test certain conditions in the IBM 7750 and to temporarily suspend the data transfer when other functions of more importance (e. g. the execution of the Service Mode or Channel Service Mode Programs) must be performed by the IBM 7750. As a result of command-signals, the proper Interface Lines become conditioned for the transfer.

The IBM 7750 cannot establish the necessary conditions for data transfer. It may send an Attention signal to the Processor indicating a certain condition - for example, the readiness for the transmission of a status message or a control message or even data. The meaning of such a signal will be determined in the specifications of each system application and it may have several meanings depending on certain operational phases or predetermined conditions. Nothing can be transmitted before the Processor responds and sends the necessary Command signal following the Attention Response. The IBM 7750 may be placed in the Out or In Mode by its program for the execution of routines implemented in the Out or In Mode Program in order to obtain higher priority under certain conditions; but in order to turn control to the Copy Mode and perform data transfer, the preceding Out or In Mode operation must now be requested by a Command from the Processor.

Commands and signals from the Processor must be recognized by the IBM 7750 program before the required actions can be performed. The four Commands, Read, Sense, Write, and Control change the mode of operation in the IBM 7750 to the Out Mode or In Mode. However, there are a number of other original, or responsive Processor signals, that must be recognized by the IBM 7750 program before, or after a data transfer operation. This can be done through the testing of the statuses of certain bit positions in the Interface Control Register. The IBM 7750 must also send program-initiated signals to the Processor, and this can be done by the programmed settings of certain bits in the Interface Control Register.

2. INTERFACE CONTROL REGISTER. The organization of the Interface Control Register is shown below.

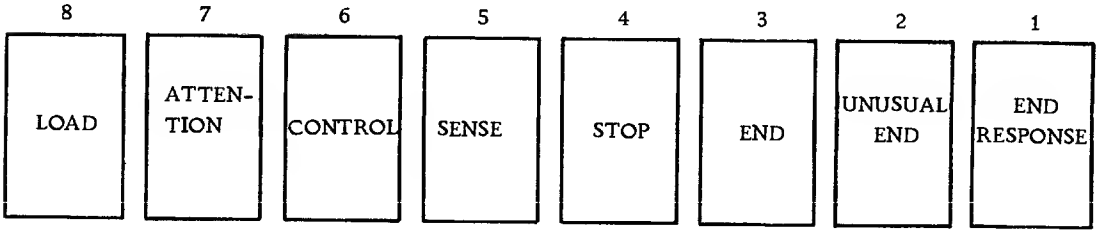


FIGURE 3: The Interface Control Register

The Load Bit (bit 8) is set to logical One for a Loading or Unloading operation. (See details in Section IX.)

The Attention Bit (bit 7) is set to logical One by the IBM 7750 program to send an Attention signal to the Processor. The program specifications must point out the conditions permitting this to be done. It is reset by the Processor's Attention Response signal. Normally, this bit will be set by the Normal Mode

Program, and provisions will be made for testing this bit, after it has been set, in the Normal Mode Program loop. A counter may be set, and if the bit is still "on" when the count has been decremented to Zero, an error message will be sent to a Terminal located at the Central Installation.

The Control Bit (bit 6) is associated with the In Mode. The machine logic may, or may not, set this bit to logical One at the same time that it sets the In Mode Bit in the Mode Request Register. This depends on sensing a Control or a Write Command signal received from the Processor, respectively. The In Mode Program must test the status of this bit position and, if the bit is "on", initiate the necessary action required by the program specifications for such a case. An example: when the Control Bit is "on", the Processor indicates that the next transmission is a control message not exceeding 30 characters; therefore, the IBM 7750 In Mode Program must set a Copy Word for only one block, and it must represent a different input area than the one used for data reception (i. e., it must be controlled by a different Limit Word). Provisions may also be made to examine the contents of this control message as soon as its transmission is terminated. If the Control Bit is off, the Processor indicates normal data transmission, and the input area must be set up as required for data transmission. The In Mode Program turns the Control Bit off.

The Sense Bit (bit 5) is associated with the Out Mode. The principles are the same as for the Control Bit in the In Mode. It is turned on by the machine logic upon sensing the Sense Command signal received from the Processor and turned off by the IBM 7750 Out Mode Program.

The Stop Bit (bit 4) is turned on by the Stop signal from the Processor whenever the Processor terminates a data transfer operation. Normally, an In operation is terminated by the Processor, and an Out operation by an equal compare of Address and Limit in the Copy word. It must be turned off by the IBM 7750 Out or In Mode Program.

The End Bit (bit 3) is turned on by the IBM 7750 Out or In Mode Program to signal a completed valid transmission to the Processor after the necessary checking procedure following the termination of a Copy Mode operation has been completed. It is turned off by the machine logic upon recognition of a response signal from the Processor.

The Unusual End Bit (bit 2) is turned on by the IBM 7750 Out or In Mode Program to signal an incomplete or invalid transmission to the Processor as a result of the checking procedure, after a terminated Copy Mode operation. For the resetting of this bit the same principles will be applied as for the End Bit. The required corrective actions must be described in the program specifications of each individual system application.

The End Response Bit (bit 1) is automatically set upon the reception of a signal from the Processor as a response for an End or Unusual End signal from the IBM 7750. This procedure is connected with the resetting of the End or Unusual End Bit.

The IBM 7750 Out Mode or In Mode Program must reset all bits in the Interface Control Register before turning the request bit off in the Mode Request Register, thus causing the machine to leave the Out or In Mode.

3. INTERFACE DATA REGISTER. Data is transmitted between the IBM 7750 and the Processor one character at a time. When transmitting to the Processor, each character is moved from the Process Storage Data Register to the Interface Data Register, and from there to the Processor via the Simplex I/O Interface. When receiving from the Processor, the order is reversed. The Interface Data Register normally is a 9 bit register; the 8 low order bit positions are used for housing data bits and the high order bit position (bit 9) for the Parity Bit. However, its size may well be adjusted to the number of bits, plus parity bit, of a character the Processor uses on a specific system (latest information).

4. COPY MODE OPERATION. Data transfer between the IBM 7750 and the Processor is performed in the Copy Mode. Two characters can be transmitted in either direction during each machine cycle (28 u) while the IBM 7750 is operating in the Copy Mode; therefore, the transmission time of one character is 14 u.

The Copy Mode is a fully automatic mode; no program is executed while the machine operates in this mode. Therefore, no program is to be written for Copy Mode. For the same reason, the Copy Mode Process Word, Copy Word for short, does not contain an Instruction counter, nor X and Y Registers. Instead, its organization shows a similar format to a Process Storage Limit Word. (See Appendix I- Chart 3). It has a 16 bit Address field, a 16 bit Limit field, an 11 bit Z Register and a Word Parity Bit. The IBM 7750 Out or In Mode Program must set the Copy Word. (See Section V - 3, 4.) The Z Register has a special function in the Copy Mode as described below.

The Copy Mode Bit in the Mode Request Register must be set by the Out Mode or In Mode Program. It is reset automatically at the termination of the Copy Mode Operation. The operation may be terminated by the Stop signal from the Processor, by an equal compare between the Address and Limit in the Copy Word, or by a time-out in Register Z.

During Copy Mode operation, the Address in the Copy Word is automatically incremented by one after the transmission of each character. An End Of Block (i. e. five Ones) test and an Address-Limit compare are performed following each increment.

Upon the recognition of an End of Block condition, the access to the Block Control Character and the replacement of the old address by the new one, including the change of the five low-order Ones to five Zeros, is fully automatic. When a new address is obtained, an Address-Limit compare is the first function the machine performs. If, at any time, an equal compare occurs, the transmission is terminated and the Copy Mode Bit is automatically turned off in the Mode Request Register. The comparison is performed after an Address increment or the replacement of the exhausted block address; therefore, the Address is always set one ahead of the last accessed Process Storage Character location. Obviously, in order not to terminate the transmission before the last desired or possible character position has been accessed, the Limit must be set one character address ahead by the IBM 7750 Program. *

At the termination of transmission, the Copy Mode Bit is automatically reset in the Mode Request Register, and control is immediately returned to the mode which requested Copy Mode (or as soon as it becomes the highest requested priority mode). The checking for successful valid transmission is done in the "requesting"- mode. (See Section V -3, 4; VI -2.)

As previously stated, the Copy Mode operation may be terminated (automatic) in one of three possible ways: recognition of an equal compare of Address and Limit in the Copy Word (normal for Out-Mode-requested transmission to the Processor); a Stop signal from the Processor (normal for In-Mode-requested transmission from the Processor); and "time-out" in Z Register.

The "time-out" method of turning control back to the Out or In Mode is the automatic cycle-number control through the contents of the Z Register in the Copy Word. The programmer may set the contents of the Z Register to any binary number up to 3777_8 when preparing the Copy Word. For the transmission of a character, the IBM 7750 machine logic issues a so-called "service request" to the Processor by setting a special hardware trigger serving this unique purpose through its output. The actual character transmission takes place after the Processor issues a "service response" which resets the trigger. If the Processor fails to do so, character transmission cannot take place, and the content of the Z Register is decremented by one in each Copy Mode machine cycle (i. e., in each 28u when the IBM 7750 is operating in the Copy Mode), until either the "service response" is received from the Processor, or the content of the Z Register is decremented to all Zeros. If, the content of the Z Register is decremented to all Zeros, the Copy Mode Bit is automatically reset in the Mode Request Register and control is turned back to the requesting mode, as was previously described. If however, the "service response" is received before "time-out", character transmission takes place, and the content of Z Register is automatically reset to 0000_8. If during the same transmission, the Processor again fails to send a "service response" to a "service request", the procedure starts again.

* The Limit must not have five one's in the low-order bits as this would prevent ever getting an Address Compare. Recognition of five one's in the Address causes it to be replaced before a compare is made.

Register Z is always decremented before the "time out" test is made. Therefore, if Register Z contains all Zeros when the decrement is performed, the result will be all Ones (i. e., the machine logic treats it as the twelve bit number, 4000_8, before the decrement; it becomes 3777_8 after).

Note: The programmer may set Register Z to any binary number lower than 3777_8 for controlling the first character transmission failure. But once such a failure occurs and it is corrected before the content of Register Z is decremented to all Zeros, the program no longer has control over the setting of Register Z. It will automatically be reset to 0000_8 after the character transmission operation is restored.

The Loading or Unloading operation is similarly performed in the Copy Mode. However, during manually initiated Loading or Unloading, the Copy Mode is automatically requested, but not by the IBM 7750 program. The procedure for Loading and Unloading operations are quite different; they are described in Section IX.

SECTION VII

MODE SELECTOR OPERATION

The IBM 7750 may, at any time, operate in only one of its six modes. The automatic procedure which determines the mode the machine should operate in, and the change, when necessary, of the operation from one mode to another is called the Mode Selector Operation.

Two hardware registers serve the purpose of the Mode Selector Operation: the Mode Request Register (MRR) and the Mode Status Register (MSR). Both are five-bit registers having one bit position for each mode except the Normal Mode. The MRR is addressable, the MSR is non-addressable.

The request for any of the five priority modes is indicated by the automatic, or programmed, setting of the corresponding bit position in the Mode Request Register to logical ONE. Therefore, Ones may be present in more than one bit position of the MRR at any time. The resetting of a bit to Zero may also happen automatically, or by the program. When there is no request for any of the five priority modes, all the bits in the MRR are Zeros. The rules for the setting of each bit are shown in Appendix IV, Chart 2.

The status of the Mode Status Register determines the mode in which the IBM 7750 is to operate. Therefore, only one of its five bit positions may contain a One bit at any given moment. If the MSR contains all Zeros, the machine is operated in the Normal Mode.

When the machine is operating in a programmed mode, the contents of the MRR and the MSR are automatically checked against each other at the end of the execution of every instruction. For a two-cycle instruction, the check is made only at the end of the second machine cycle. During the Copy Mode operation, the check is performed at the end of each machine cycle. If the check shows that the machine is operating in other than the highest requested mode, the MSR is updated to the highest requested mode (i. e., the bit corresponding to the highest requested mode is turned on and the bit, if any, which was "on" at the time of the checking is turned off). If the MRR contains five Zeros, the MSR will also be reset to five Zeros if any one of the bits of MSR was "on" at time of checking. The machine logic examines the status of the MSR; if there is no change in status, the operation mode does not change; if there is a change in status, the Process Word of the new mode is accessed and read out from Control Storage into the Control Storage Data Register. By this means, control is turned over to the newly selected mode with the loss of one machine cycle. (See Appendix IV, Chart 3 for Mode Selector Operation.)

In some cases, when an operation must be completed without allowing the program of another mode to operate on the same data or location, the program may delay the change of mode for the cycles necessary to complete the progressing operation. For this purpose, those instructions required to perform and complete the desired operation, without interruption, must be flagged to Prevent Mode Change - except for the last instruction of the routine, after which the mode of operation may be changed without any danger of such undesirable interference.

SECTION VIII

TRAFFIC CONTROL PROGRAM: POLLING, RECEIVING, SENDING.

A "TELE-PROCESSING" Network consists of a number of Terminals located in remote and/or local offices, connected to the proper facilities of a Central Installation by Communication Channels. The connection may be direct, or indirect through intermediate multiplexing devices called Interchanges or Exchanges. Various combinations of different types of Terminals, Communication Channels, and connection methods may be incorporated into a Network. Transmission speed requirements (i. e., bit rates per second), traffic volumes, and geographical locations are the most important factors to be considered when designing a network layout.

Reliable and efficient operation, projected to the expected maximum (i. e., peak) traffic load, determines the limits for the highest possible utilization of the channels and equipment. Special requirements by the user may be considered within these boundaries only.

In order to meet these requirements, and establish and maintain a system of high reliability and efficiency, the programming personnel must share the responsibilities with the technical personnel. The technical requirements and limitations of the equipment must be satisfied, and rules must be established by the network layout design; the proper utilization of the available facilities and the assurance of adequate traffic distribution are programmed functions. The IBM 7750 controls the entire operation of the Network, and the program written for the IBM 7750 commands the machine in performing this important task, and, in addition, many other functions. Programmed methods called POLLING, RECEIVING, and SENDING are used to direct and control the entire traffic of a "TELE-PROCESSING" Network. The organized combination of these three methods may be referred to as the "Traffic Control Program". It is not a separate program, but adequately "built in" into the Normal Mode Program complex of the IBM 7750. The sometimes-used statement that Polling, or the Polling Program, "controls the entire traffic" of a Network is insufficient. However, Polling has a "more independent" form of programming than Receiving, or even Sending; for that reason, it has often been recognized as the dominant method of traffic control.

POLLING is the organized method of allocating Communication Channels to Terminals for incoming traffic. In many "TELE-PROCESSING" Systems, more than one Terminal will be connected to all or to some of the Communication Channels. In order to use the channel, a Terminal will be given "permission" by the IBM 7750 to do so. This "permission", depending on the type of facilities used, may be a unique signal or a short control message - a "polling message" - transmitted to the Terminal. The characters of a polling message have to consist of predetermined bit patterns to provide a means for recognition through wired-in

"logic" by "contact units" (i.e., Stunt-Boxes, etc.) connecting the Terminal units to the Communication Channels.

If high-speed channels connect multiplexing devices to the IBM 7750, and each multiplexing device services a number of low-speed channels and a number of Terminals through each of these low-speed channels, the IBM 7750 controls the traffic directly to and from the multiplexing devices. Each such device controls the collection and distribution of data from and to its Terminals. This is done on a predetermined, but changeable, polling schedule in input, and through the examination of certain "destination" characters in output operations. But, even in such applications, the IBM 7750 has indirect control over the entire traffic down to the Terminals. The input traffic may be controlled by changing the polling sequence of the multiplexing device by special signals. The output traffic is directed by the insertion of proper destination characters in the IBM 7750 output areas.

When the IBM 7750 polls a Terminal, the recognition of special polling signals or characters will initiate a response (i.e., a message or a "no message" indication) by the individual Terminal. When certain types of transmission facilities are incorporated into a system, some other functional characters may be implemented into a polling message besides the "contact-establishing" characters; for example, characters causing the turning-on of a light on the Terminal set, a carriage return, line feed, or some other function. If a Terminal is a typing unit, the characters used in a polling message are not printed. Similarly, on other type of Terminals, the polling signals or characters normally will not be recorded; their functions are "contact-establishing" and "conditioning" necessary for the Terminal to be able to transmit information.

The above description clears the concept of Polling, and the purpose and construction of polling signals, or messages, to the point where it can be seen that a Terminal or Interchange is actually polled in the sense of sending out recognition-seeking "contact-establishing" and "conditioning" characters. Through them, a Communication Channel is allocated to a transmission unit for the purpose of transmitting to the IBM 7750 one or more messages, or a part of a message, depending on the specific application. The polled unit may respond with either a message, or usually an automatically generated signal (character) indicating that it has no message to send. In cases when only one Terminal unit is connected to the IBM 7750 by one (usually high-speed) channel, normally a signal exchange is performed preceding a transmission, and each received message is acknowledged by the Terminal with a signal. Consequently, polling and similar signal exchanges enable the IBM 7750 to maintain an awareness of the status of the Network, because a failure to respond to a polling signal or message with either a message or a "no message" signal, or to transmit some other established signal, would indicate an inoperative unit and corrective measures may be initiated.

Polling also serves the important function of allowing the IBM 7750 to inhibit the receipt of terminal data. Normally, the program would not "invite" the receipt of a terminal data message until it had ascertained that sufficient buffer storage was available to contain the message. Failure of the Processor, inordinate traffic loads, etc. may overload the buffer storage that is available to the IBM 7750. The inhibition of polling for short periods of time will provide for these contingencies.

The organization of Polling must be planned independently for each Communication Channel. Some transmission units on a channel may produce considerably more input for the IBM 7750 than others. The balance of the traffic would be disrupted if Terminals transmitting different amounts of data were polled equally. Therefore, after careful consideration of the traffic volumes of each Terminal relative to the other Terminals on the same channel, a Polling Sequence must be scheduled. At least one Polling Chain containing successive identifying characters assigned to each Terminal must be constructed for each channel. The Terminal Identifier will appear in the string of identifying characters (i.e., Polling Sequence) as frequently as it is desired to poll the specific Terminal in proportion to the total number of identifying characters in the Polling Chain. A Limit Word controls each such chain, which may be located in one or more blocks, as necessary. The last block of the chain should be linked back to the first block, thus forming a closed chain. Each time the program leads to the execution of the polling routine for a channel, the next identifier in sequence will be obtained from the Polling Chain through the controlling Limit Word; then the proper polling characters will be accessed through the identifying character and used in the forthcoming Polling procedure. The Address in the Polling-Chain-controlling Limit Word is incremented by one each time after an identifier has been obtained, thus readying it to access the next identifier in sequence. Whenever the last character in a block has been obtained, the Address in the controlling Limit Word is changed to the first character address of the next block in the Polling Chain; finally, after the entire contents of the chain have been exhausted, the Address in the Limit Word is set to the very first character address of the Polling Chain. The automatic End Of Block recognition feature is used for the interblock address changes through the use of the proper "indirect-increment-skip" instructions. Often, more than one Polling Chain is needed for a channel, because during certain time periods of the day, a different one is required. For example, some branch offices may close for certain hours during the day, and some may work longer hours than others, or may even operate on a 24-hour basis. An example is given below of a general type of Polling procedure.

Example: Nine Terminals, A through I, are connected to a channel. Due to the relative terminal activity, it is desired to poll the Terminals as follows:

Terminal Identifier	No. of times polled in one sequence
A, B, C	7 (each)
D	6
E	5
F	4
G	3
H	2
I	1

The Terminal Identifiers representing the Polling Sequence are stored in a closed chain in Process Storage as shown in FIGURE 4.

POLL (1st Block) =
XXXXXXXXXXX00000

*NOTE: The sequence is arranged so that the last Identifier (I) is located in character position 31 of the second block. The blocks need not be in sequential order.

CONT (2nd Block) =
YYYYYYYYYYY00000

A		B	C	D
* A		B	C	A
B		C	D	A
B		C	D	BCC Y---Y
E		A	B	C
D		E	F	A
B		C	D	E
F		G	A	B
C		D	E	F
G		H	A	B
C		D	E	F
G		H	I	BCC 0-----0

FIGURE 4: Polling Chain in Process Storage

The program to obtain the Terminal Identifiers is shown in FIGURE 6. The following labels (Names) are used:

SEQNCE	Location of instruction used to obtain Terminal Identifiers
POLL	Label of first block
CONT	Label of second block
POLPRE	Start of Polling preparation routine
EOB	Start of End Of Block routine
ZEROS	Start of End Of Sequence routine
POLLWD	Limit Word controlling Identifiers in the closed chain. The organization of this LWD is shown in FIGURE 5.

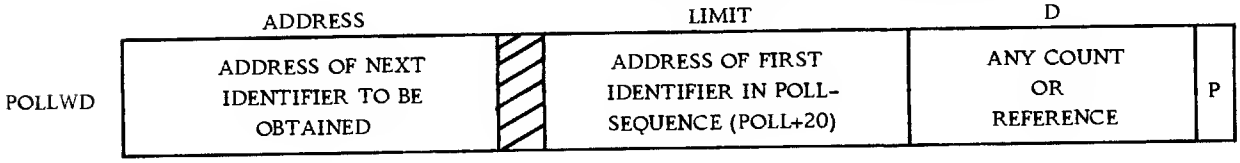


FIGURE 5: Polling Chain Controlling Limit Word

NAME	OP CODE		R	S		F		ADDRESS, ML	COMMENTS
SEQNCE	LOI*		Z	1	1			POLLWD	OBTAIN IDENTIFIER
	BRA							EOB	BRANCH WHEN EOB CONDITION
POLPRE	Routine for Polling preparation								
	BRA							EXIT	LAST INSTR. OF POLLING PREP. ROUTINE
EOB	LOD*		Y	1	1			POLLWD	OBTAIN BCC
	BRZ		Y	1	1			ZEROS	TEST BCC;BRANCH IF ALL ZEROS
	LOI							POLLWD	FORCES 5 LOW ORDER ZEROS IN ADDR. FIELD
	UNL		Y	1	1			POLLWD	CONT TO 11 HI ORDER POS. OF ADDR. FIELD
	BRA							POLPRE	PROCESS IDENT. OBTAINED FR. CHAR. POS. 31
ZEROS	GTL							POLLWD	OBTAIN CONSTANT: POLL + 20
	PTA							POLLWD	RESET ADDR. FIELD TO POLL + 20
	BRA							POLPRE	PROCESS IDENT. OBTAINED FR. CHAR. POS. 31

FIGURE 6: Polling Sequence Program

The Program shown in FIGURE 6 would be the same regardless of the number of blocks required to store the Polling Sequence. The all-zero test character should, of course, be stored in the BCC of the last block of the chain.

Polling is considerably different on half-duplex than on full-duplex channels.

Polling on half-duplex channels may be done at the end of an outgoing transmission, or independently, by sending out only a polling message.

Polling at the end of an outgoing transmission is done by storing the polling characters in the output area, after the last character of the outgoing message. Since the channel must be placed into Receive Status immediately after the polling characters have been sent out, the Status Change feature must be utilized by storing the Status Change Character next to the last polling character, with or without a following DCC as required. (See Section IV-4, D).

Polling independently should be done when the program does not find a prepared message in the Channel's output area, or when it is desired to increase the input activity for better utilization. Some applications may even require a constantly higher input operation than output if, for example, the incoming messages are much longer than the outgoing messages and/or only a certain percentage of the incoming messages require a response.

Polling on full-duplex channels may be done in three different ways: at the end of an outgoing message transmission, independently, or during an outgoing message transmission by inserting the polling message between two consecutive characters of the outgoing message through manipulation of the character addresses.

Polling at the end of an outgoing message in most applications cannot be programmed as a regular method, as can be done on half-duplex channels, because the Receive line of the channel may not be free by the time a Terminal is polled this way. However, it can be done if the Receive line is free when the transmission of the outgoing message begins, or if the shortest outgoing message is equal to or longer than the longest incoming message, should such specifications ever occur in an application.

Polling independently can be performed when the program detects the termination of the input operation by finding the End of Transmission, or End of Message character, or a "no message"-indicating character from the last polled Terminal in the input area; and the following check on the Send Channel Word shows that the last sending operation has already been terminated (i. e., Send-Receive bit position contains a Zero). The program then branches to the regular polling routine, which prepares the transmission of the polling character, or message, as an independent action.

Polling during an outgoing message transmission is performed if the Receive line must be utilized as soon as the status of the input area indicates its availability, and the following check on the Send Channel Word shows that an output operation is in progress on the Send line (i. e., The Send-Receive bit position contains a One bit). In such cases, the status of the output operation in progress must be checked by the program by further examination of the contents of the Send Channel Word (i. e., Bit Counter, Delay Bit if Start-Stop, Assembly Area if Delay Bit is "on") in order to determine whether the polling message can be inserted without risking an unwanted interference that may result in the loss of an output character. The polling characters must be kept as constants in the last data character positions of a block (i. e., the last one must have an address ending in 11110 binary bits). The next step will be to replace the Address of the next output character in the Send Channel Word with the Address of the first polling character and place the block address (i. e., the eleven high-order bits) removed from the CWD into the Block Control Character (i. e., ending in 11111 binary bits) of the block containing the polling message. At the same time, the five low-order bits of the Address removed from the CWD, representing the word and character address, must be saved in a Process Storage "save location". Consequently, after the last (i. e., high-order) bit of the output character (located in the Assembly Area of the Send Channel Word when this address change in the CWD is performed) has been sent out, the first character of the polling message will be loaded into the Assembly Area during Character Interrupt, and, from here on, the polling characters will be transmitted out one-by-one. After the last polling character has been obtained from Process Storage, Channel Service Mode is requested. In the Channel Service Mode Program, either a special Send routine must be implemented, in addition to the regular Send routine, for such cases, or the Send routine should be programmed to obtain the five low-order bits of the new address from the "save location" for all cases. When executing the Send routine, the block address must be obtained from the Block Control Character and the five low-order bits from the "save location". A convenient arrangement for "save locations" would be a 128 position character table indexed by the 7 bit Control Storage Addresses of the Channel Words. Whenever a Channel Word is not affected by a polling message insertion, all Zeros must be kept in the corresponding table entry. It must also be set to Zero after a saved item is replaced in the CWD. By this method, the removed character address will be replaced in the Address field of the CWD, if such removal took place previously and the first character address of the next output block if no previous removal took place. An other than all Zero table entry may also serve as an indicator for the Channel Service Mode Program, so that the block which caused the request for the Channel Service Mode (i. e., the address of which is located in the affected CWD) will not be released because it contains polling characters. If an all Zero table entry is accessed, the block should be released; therefore,

an address with five low-order Zeros cannot be replaced in the CWD with the first polling character address. The program must test for this condition before the address change.

There are several other possible methods to handle Polling during an outgoing transmission. The instructions for checking the contents of a Send: CWD and changing the Address in it for a Polling Character address must be flagged to prevent mode change; otherwise unwanted complications may arise sooner or later.

RECEIVING operations presuppose the proper status of the incoming channels. Preparations such as Adapter Synchronization, or other settings (e.g., preparation for the recognition of the Action Delay Character when necessary), must be done by manipulating the Channel Word. However, the Receive line of a full-duplex channel may be kept in Receive status throughout the operation of the system, without any programmed intervention. On half-duplex channels, the Status Change feature and the Action Delay feature (if used) may absorb the preparation entirely, and very little, or no further, programmed intervention would be necessary. Functions that must constantly be taken care of by a separate Receiving program routine are, the detection of the lack of response from a polled transmission unit and the initiation of the necessary action to be taken. If such a case occurs, the desired actions (i. e., the repetition of Polling and/or the initiation of corrective actions) may diminish in the Polling or Sending (i. e., sending out a repeated polling message), and in the error routines. If the check indicates that the response has been received from the polled Terminal, the rest of the job (i. e., the message processing) will be turned over to the proper processing routine of the Normal Mode Program.

SENDING. There is a similarity between Polling and the beginning of a Sending operation. The Terminal(s) or Interchange(s) to which the outgoing information is directed must be selected by the use of "contact-establishing" signals, or characters, frequently called Call Directing Codes (CDC). These are simply placed by the program in the header of the outgoing information instead of forming a separate control message. All or any combination of receiving units (i. e., usually send-receive units) sharing the same Communication Channel may be directed to receive a message by the addition of the required Call Directing Codes. The same message may be sent out on all channels to all, or any number of selected receiving units. Consequently, the outgoing messages may be individual, multi-or-group-addressed, and broadcast messages. Some types of receiving units transmit an automatically-generated "acknowledgement" signal upon the receipt of a message in order to provide the IBM 7750 with a means for checking the awareness of the Network.

The IBM 7750 needs no permission to send out messages (except to the CPU).

The only limitation is that the Communication Channel must be free (i.e., not busy) and the receiving unit operative. Both such transmission-limiting factors would occasion queueing (i.e., building up a string of messages for a channel).

Sending operations normally require more programmed functions than do Receiving operations. The output area of a half-duplex channel and of the sending line of a full-duplex channel must be regularly examined, and the transmission of the prepared messages in the output queue must be initiated by the Sending program routine. Since a channel cannot be "kept open" in Send Status,as it can be in Receive Status, Adapter Synchronization will be needed quite frequently, preceded by the setting of the Send-Receive Bit to Send (and the setting of the Not Hold Bit and resetting of the Delay Bit in Start-Stop character operations if the Channel has been inactive), since it is set to Receive each time an outgoing transmission is terminated by the application of the Status Change feature. The prepared messages in the output area are controlled by Limit Words. A Master Limit Word controls the string of such output area controlling Limit Words, each normally representing a message. The Sending routine then releases the messages one-by-one through this two-level Limit Word manipulation.

The relationship between the introduced functions can be summarized, in a general way, as follows:

> Polling is closely associated with the necessary preparations for Receiving. Furthermore, the checking of the received data indicates whether or not the condition for another Polling is established (i.e., whether the channel is free or occupied). No Terminal can be polled on a Communication Channel until the program has determined the termination of the previous input operation on that channel. The lack of response from the polled Terminal must also be detected by Receiving program routines. Neither can Polling be separated from Sending. Polling is nothing else but sending out polling characters or polling messages. When a system contains half-duplex communication Channels, and most systems are likely to have such channels, no Sending activity may be started on a channel after Polling,until the initiated (i.e., by Polling) Receiving activity has been terminated; and no Polling can be done until the Sending activity has been terminated. On full-duplex channels, in order to utilize the Receiving line efficiently, Polling must sometimes be done during Sending activity by actually interrupting the continuous transmission of an outgoing information unit. Sending and Receiving operations are inseparable for half-duplex channels; the initiation of one depends on the termination of the other.

Conclusion: The function of the Traffic Control Program is to prepare, initiate and control incoming and outgoing data transmission on the entire "TELE-PROCESSING" System throughout the period of operation, in an organized fashion

in order to assure the highest utilization of the Communication Channels, the quickest possible service for the Terminals, and the continuous input-output data flow for the Center. Therefore, Polling, Receiving, and Sending must be carefully implemented in a well-coordinated manner into a Traffic Control Program by scaling the expected incoming and outgoing traffic volume figures for each Terminal on the system, and by taking into consideration many other functional and technical factors.

SECTION IX

LOADING-UNLOADING

Loading or Unloading the IBM 7750 is performed through its associated computer under the control of the Load Trigger, which is located in bit position 8 of the Interface Control Register. This Load Bit can be set by the IBM 7750 program, or manually from the Operator's Panel by pressing the Load Button. However, the machine must be in Halt status when the Load Button is pressed.

During Loading or Unloading, complete 48 bit words are transmitted between the Processor and the Process Storage of the IBM 7750 in 16 groups of 3 bits. A special hardware counter controls the transmission of the 16 groups, and it is reset automatically after a full word has been transmitted. When Loading, each group of 3 bits will be placed in the three low-order bit positions of the Interface Data Register and from there into three consecutive bit positions of the Process Storage Data Register starting with the three high-order bit positions (47-46-45), continuing sequentially, and ending in the two low order plus the Parity bit positions (2-1-P). After the assembly of a full word, an odd parity check is made and the word is written into Process Storage. The transmitted words will be stored in consecutive Process Storage word locations. When Unloading, each consecutive word is read out from Process Storage into the Process Storage Data Register, a parity check is made, then each group of three bits will be moved into the three low-order bit positions of the Interface Data Register, and from there to the Processor. The 3 bit groups are transmitted in the same sequential order as in Loading, beginning with the three high-order bit positions (47-46-45) of the Process Storage Data Register.

If Loading or Unloading is initiated by the program, the Address in the Copy Word must be set to the first word location, and the Limit to one location higher (i. e., to the "A" character address of the next word) than the location where the last word is to be stored in, or obtained from, Process Storage. The operation may be terminated by a Stop signal from the Processor, by an equal Address-Limit compare in the Copy Word, or by time-out in Register Z (i. e., when its contents have been decremented to all Zeros). Normally, the Loading operation is terminated by the Stop signal and the Unloading by an equal compare.

During manually-initiated Loading or Unloading, the first word will be stored in, or obtained from, the Process Storage word location whose address is all Zeros (i. e. the first Process Storage word location). A Stop signal from the

Processor must terminate the operation because the Copy Word is reset to Zero the first time it is read out of Control Storage in order to access the first word location in Process Storage through its all Zero Address. Consequently, the Limit field will also contain all Zeros, and for that reason, an equal compare cannot terminate the operation. However, a time-out in Register Z may also terminate the transmission.

During Loading or Unloading (i. e., while the Load Bit is "on" in the Interface Control Register), the End Of Block test is not performed by the machine because all consecutive word locations must be accessed in the affected Process Storage area without a gap. The Load Bit must be reset by the IBM 7750 program after the operation has been terminated.

Since data transmission between the Processor and the IBM 7750 cannot be performed unless the Processor establishes the necessary conditions in the Simplex Interface, neither Loading nor Unloading can be accomplished without the Processor's preliminary Write, or Read, Command.

For the initial Loading of the machine, the Load Bit must be set manually from the Operator's Panel. Pressing the Load Button will also set other hardware triggers, such as, the Attention Bit in the Interface Control Register, the Manual Load Trigger, and the Mode Change Trigger (i. e., the trigger to inhibit functions during the cycle in which the mode change is performed). The Attention Bit will raise the Attention line, signaling the readiness of the IBM 7750 for the Loading operation to the Processor. The Processor resets the Attention Bit and sets the In Mode Bit in the Mode Request Register. The In Mode Process Word will automatically be reset to Zero when it is first read out. However, during the time when the Manual Load Trigger is "on", and a Write Command is not received from the Processor, the Mode Change Trigger will be kept "on", thus preventing the machine from executing instructions. This is necessary because an Op-Code Parity Error on the previously reset words would stop the machine immediately. Normally, both Storages and all registers and triggers are cleared (i. e., reset to Zero), before initial Loading, from the Operator's Panel by the Clear Switch. The Manual Load Trigger will be kept "on" until the first word has been loaded. In addition to the described function, the "on" status of the Manual Load Trigger will cause the Copy Mode Bit in the Mode Request Register to be turned on immediately after the Write Command has been received, causing the Mode Change Trigger to be reset; from this point on, no further protection is necessary to prevent instruction-execution because the Copy Mode has no program.

Control Storage must be loaded by the IBM 7750 program. Since the In Mode Process Word is automatically reset to Zero when first read out, the In Mode may conveniently be used to load Control Storage after the first Loading

operation has been terminated. Since the content of the Instruction Counter in the In Mode Process Word contains all Zeros, the first instruction of the Control Storage loading routine must be the first word transmitted to Process Storage during Loading. As soon as the necessary instructions and word-images, prepared for Control Storage, are stored in Process Storage, the Loading operation may be stopped ; the Control Storage loading routine can be executed. Loading may then be started again beginning with the first Process Storage word location in order to overlap the already executed instructions and utilized "constants" (i. e., word-images), since they will no longer be needed.

Manually initiated Unloading is performed on the same basic principles as Loading, but, of course, a Read Command is received from the Processor, and the Out Mode Bit is set in the MRR. However, contrary to the procedure followed for the In Mode Process Word during Loading, the Out Mode Process Word will not be reset to Zero.

SECTION X

THE IBM 7750 INSTRUCTIONS

1. DESCRIPTION

The available instruction set is composed of the basic instructions and their variations resulting from the application of modifying micro-codes. There are 12 basic instruction types which, through modification by micro-codes, are increased to 77 actual instructions. Each instruction may be used in any Mode Program.

The execution time of an instruction may be one machine cycle - 28u - or two machine cycles - 56u; therefore, instructions are described as One-Cycle or Two-Cycle Instructions. Those using indirect addressing (except Branch instructions) and all Storage-to-Storage instructions are Two-Cycle Instructions; all others require only one cycle for their execution.

Referring to their functions, instructions are divided into four main categories. These are:

TYPE I	- CH	-	Character Manipulating Instructions
TYPE II	- AL	-	Address and Limit Moving Instructions
TYPE III	- SS	-	Storage-to-Storage Data Transfer Instructions
TYPE IV	- CT	-	Control Instructions

Every instruction has a 10 bit binary Operation Code. References to these actual machine language Op-Codes are given in octal numbers. For the convenience of programming personnel, every Op-Code is also expressed in mnemonic characters, referred to as mnemonic Op-Codes. Programs to be assembled by the IBM 7750 Assembly Program must be written in symbolic language (i. e., using mnemonic Op-Codes).

The mnemonic Operation Codes have been constructed in a highly efficient, practical,and logical manner. The first symbol of any mnemonic Op-Code has the same unmistakable meaning. Therefore, the programmer, after a very short period of coding, will practically memorize all the instructions.

The second, third and/or fourth symbols of a mnemonic Op-Code may represent a micro-code,or combination of micro-codes. A micro-code may be considered as a change in the bit pattern of a basic instruction which causes the machine logic to execute this instruction in a modified manner,and/or to perform additional function(s) in addition to the basic function. The use of a micro-code may be represented in a mnemonic

Op-Code by the letter C in the second or third symbol position, I in the third position, * in the fourth position, or I* in the third and fourth positions.

An explanation of the uses of micro-codes represented by these symbols is shown under C, I, *, and I* in Chapter 2 of this section.

The meanings of the first letters of the Operation Codes are shown below. As can be seen, they can be easily memorized.

First Letter of Mnemonic Operation Code	Function of Instruction
A	AND
B	BRANCH
C	COMPARE
G	GET (Into OAR)
I	INCLUSIVE OR
L	LOAD
M	MOVE WORD
O	OR TO PROCESS STORAGE (Inclusive Or)
P	PUT (From OAR)
T	TRANSMIT ADDRESS
U	UNLOAD
X	EXCLUSIVE OR

The instructions are introduced in this Guide in convenient table formats. Instructions having identical first letters in their mnemonic Op-Codes are located in the same Table. The descriptive name, mnemonic and octal Op-Codes, execution cycles,and all functional details are given for each instruction. In addition, for the convenience of the user, an identification number from 1 to 77 is assigned to each instruction.

2. TERMINOLOGY: EXPRESSIONS, SYMBOLS, ABBREVIATIONS

The symbols, abbreviations, and names used in the description of the instructions(including the basic instruction modifying micro-codes,and the single and combined bit positions (fields) of the Instruction Word)and their functions are explained in this chapter. In addition, the AND-OR logic is explained.

IWD = INSTRUCTION WORD. Each instruction is stored in a full word in Process Storage. An Instruction Word must contain an

Operation Code and all the additional control information the machine logic needs in order to perform the desired function(s) on the contents of the specified Storage location(s), hardware register, and Instruction Counter, as required. The organization of the Instruction Word is shown in Appendix II-Chart 3.

OP-CODE = OPERATION CODE. Each instruction has a unique 10 bit binary Operation Code to specify the desired function(s) to be performed by the machine logic. It is located in bit positions 10-1 of the Instruction Word.

P (bit 11) = OP-CODE PARITY BIT. Each Instruction Word has an Op-Code Parity Bit (bit 11) in addition to the Word Parity Bit. This bit position must contain either a Zero or a One bit to establish an odd parity on the 10 bits of the Op-Code and the two Flag Bits (bits 13-12). This parity bit is generated by the Assembly Program and checked in the Instruction Register before an instruction is decoded. It is never regenerated by the machine logic, as a Word Parity is before a word is written into, or written back from, a Data Register to one of the Storages; therefore, the program must take care of the proper Op-Code parity Bit setting when changing anything in the Op-Code field, or in the Flag fields.

W = ADDRESS (bits 47-32) in the Instruction Word. It always represents a Process Storage location. Everything written in the Variable field on the coding line of an instruction beginning with Column 24 up to the first comma or blank, whichever comes first, will be interpreted or computed by the Assembly Program and stored in the Address field (bits 47-32) of the Instruction Word as W. Consequently, for direct unmodified CH instructions, W is the address of a character in Process Storage to be manipulated. In the execution of all other types (AL - SS - CT) of direct, unmodified instructions, the two low-order bits of W (bits 33-32) are ignored and the high-order bits (bits 47-34) address the word to be used in executing the instruction.

W_e = EFFECTIVE ADDRESS. This is the address of a Process Storage location. It is formed in Process Storage Address Register #1 (PSAR1). The Address (W) in the Instruction Word is moved by the machine logic into Process Storage Address Register #2 (PSAR2), and from there into PSAR1 when it is to be used. If no address modification (by ML) is required, W is simply transferred from PSAR2 to PSAR1. If address modification is required, L number of low-order bits from Register M are moved into the

low-order bit positions, and 16 minus L high-order bits of PSAR2 into the high-order bit positions of PSAR1 as a parallel operation. The address in PSAR1 is referred to as W_e. W_e = W if there is no address modification by ML. Otherwise, it is W modified by ML. Depending on the type of instruction, W_e may refer to a character, to a word, or, when indirect addressing is used, to a Limit Word containing the indirect address (W*) of a character or word. All instructions may be modified by ML with the exception of the following CT instructions: BRA, BRZ, BRZ*, BRO, BRO*, and BRT.

W* = ADDRESS IN PROCESS STORAGE WORD AT LOCATION W_e. In most cases, this is an INDIRECT ADDRESS (i.e., the address of a character or a word when indirect addressing is used). It is located in bit positions 47-32 of the Process Storage Word addressed by W_e. If it refers to a word as an indirect address, the two low-order bit positions are ignored by the machine logic and the desired word is accessed through the 14 high-order bits of the 16 bit W*.

R = REGISTER. Each of the 7 addressable registers (i.e., Registers X, Y, and Z which are located in the Process Word of each of the programmed modes and the four addressable hardware registers) has a binary identification number. An instruction which has to use, or change the contents of a register during data manipulation, must refer to this number. Column 18 of the coding sheet (identified by R) may contain either the identification number in decimal, or its corresponding letter. Bit positions 29-27 of the Instruction Word will contain the identification number in binary. The identification numbers and letters are as follows: 0 (000) = NO REGISTER; 1 (001) = X; 2 (010) = Y; 3 (011) = Z; 4 (100) = C (CSR); 5 (101) = D (IFDR); 6 (110) = I (IFCR); and 7 (111) = M (MRR). See Appendix IV-Chart 4 for Registers.

S = SIZE. The number of low-order bits of Register R affected by the instruction are defined by Size. Columns 19 and 20 of the coding sheet (identified by S) must contain the decimal representation of the number of bits. Bit positions 26-23 of the Instruction Word will contain its binary equivalent. S may range from 0 to 11_{10}.

R and S = COMBINED FIELD FOR ADDRESSING A CONTROL STORAGE WORD. When executing an SS instruction, the machine logic interprets these two fields, bit positions 29-23 in the Instruction Word, as one seven bit field containing the address of a Control

Storage Word. RS may range from 0 to 127_{10}. In such a case, the binary bits in the S field, which are part of the 7 bit binary address, may show a higher number than 11_{10}. The CS address on the coding line must be placed in the RS columns as an actual one, two,or three-digit number, with the low-order decimal digit in column 20. However, when an SS instruction is implemented in the Channel Service Mode Program, no Control Storage address need be given, because the contents of the seven low-order bits in the Channel Service Register will be used by the machine logic to access a Control Storage Word.

M = MODIFICATION REGISTER. Any number of bits from each of the seven addressable registers may be used for address modification. On the coding line, the identifying letter of the selected register must follow the last character of the Process Storage address, separated by a comma, in the Variable field. A number may not be used to define an M Register. Bit positions 20-18 of the Instruction Word will contain the binary identification number of the selected M Register. The binary identification numbers and their corresponding letters are shown under the reference R = REGISTER.

L = LENGTH. The number of low-order bits of Register M to be used for address modification is defined by Length. The decimal number representing Length must immediately follow the Modification Register identification letter on the coding line. Bit positions 17-14 of the Instruction Word will contain its binary equivalent. L may range from 0 to 11_{10}.

OAR = OPERATIONAL ADDRESS REGISTER. The 11 bit positions of Register Y and the 5 low-order bit positions of Register X in the Process Word of any of the five programmed modes are interpreted by the machine logic as one 16 bit register when executing an AL instruction. The Y portion of the OAR will contain the 11 high-order bits and the X portion the 5 low-order bits of the 16 bit Address or Limit moved into the OAR from a Process Storage Word,or vice versa.

DZR FLAG = DECREMENT Z REGISTER FLAG. A One bit in bit position 12 of any Instruction Word indicates a command for decrementing the contents of Register Z by one. The decrement will take place before the execution of the instruction. On the coding line, a number 1 in column 22 (identified by F) represents this command.

PMC
FLAG = PREVENT MODE CHANGE FLAG. A One bit in bit position 13 of the Instruction Word indicates a command for prevent mode change. After the execution of an instruction so flagged, no mode change can take place regardless of the contents of the Mode Request Register. On the coding line, a number 2 in column 22 (identified by F) represents this command.

FLAGS = COMBINED USAGE OF BOTH FLAGS. A One bit in both bit positions 12 and 13 of an Instruction Word indicates a command for both decrement Z Register and prevent mode change. On the coding line, a number 3 in column 22 (identified by F) represents this double command.

* = MICRO-CODE SYMBOL FOR INDIRECT ADDRESSING. If the fourth character of a mnemonic Operation Code is an *, the instruction uses indirect addressing. The indirect address, W*, is located in the Address field of the Process Storage word at location W_e, and the data to be operated on, or the word to be accessed in Process Storage for the execution of the instruction, is addressed by W*.

Only instructions using indirect addressing have four character Op-Codes, the fourth character of which is always an *. All instructions not using indirect addressing have three character Op-Codes, every character of which is alphabetic.

* = PROCESS STORAGE LOCATION OF THE INSTRUCTION. If an * is used in column 24 of the coding line, it refers to the Process Storage location address of the same instruction in which it is used. It may be adjusted (e.g., * + 4, * - 3, etc.) and/or may be modified by ML.

* = REFERENCE TO COMMENTS. If an * is placed in column 6 of the coding line, everything written on the same line after the asterisk will be interpreted as comments. It will be printed on the Assembly Listing, but does not affect the program.

I = MICRO-CODE SYMBOL FOR ADDRESS INCREMENT. If the last alphabetic character (i.e. the third character) of an Operation Code is an I, the contents of the Address field (i.e., W*) at Process Storage location W_e are incremented by one after the execution of the instruction. The entire 16 bit address (bits 47-32) is incremented by one, whether or not this address is used as an indirect address; and if it is used as an indirect address, whether the instruction uses the address to refer to a character, or to a word.

I* = MICRO-CODE SYMBOLS FOR INDIRECT ADDRESSING AND ADDRESS INCREMENT, RESULTING IN A SKIP-TYPE INSTRUCTION. If the last two characters of a four character Operation Code are I*, the instruction is executed as an instruction using indirect addressing; and W* is incremented by one after the execution of the instruction. In addition, after W* has been incremented, its five low-order bits (bits 5-1) are automatically tested for all Ones. If all Ones are found, (i.e., an End Of Block condition is detected) the next instruction in sequence will be executed. If the five low-order bits are not all Ones, the next instruction is skipped, and the second instruction in sequence will be executed. In addition to the instructions having I* in their mnemonic Op-codes, the only two other skip-type instructions in the entire instruction set are the CAL and CAI instructions, identified by numbers 76 and 77 respectively.

C = MICRO-CODE SYMBOL FOR COMPLEMENT MANIPULATIONS. If the second or third character of the mnemonic Op-Code of a CH instruction is a C, the One's complement of the contents of one of the two data locations will be used in the character manipulation instead of the original bit pattern. The One's complement of a Zero bit is a One, and of a One bit is a Zero. Example:

Bit Pattern:	101100
One's Complement:	010011

AND = A LOGICAL OPERATION during which two binary bits are compared and a One recorded as a result if both compared bits are ONES; otherwise, the result is a ZERO. Example:

Bit Pattern #1:	0011
Bit Pattern #2:	0101
Result of AND operation	0001

INCLUSIVE OR = A LOGICAL OPERATION during which two binary bits are compared and a Zero recorded as a result if both compared bits are ZEROS; otherwise, the result is a ONE. Example:

Bit Pattern #1:	0011
Bit Pattern #2	0101
Result of INCLUSIVE OR operation	0111

EXCLU-
SIVE OR = A LOGICAL OPERATION during which two binary bits are compared and a ZERO recorded as a result if both compared bits are identical; otherwise, the result is a ONE. Example:

Bit Pattern #1:	0011
Bit Pattern #2	0101
Result of EXCLUSIVE OR operation	0110

3. THE ASSEMBLED INSTRUCTION WORD

The instructions are assembled by another computer, such as an IBM 1401, and loaded into their Process Storage locations. Each Instruction Word occupies one 48 bit word. Examples of instructions as they appear first on a coding form and then as assembled Instruction Words in their Process Storage locations are illustrated in Figure 7. Three different examples are shown. They are: (a) an instruction with a straight-forward actual operational address, (b) an instruction with an adjusted address, and (c) an instruction with an address modified by ML. Each example consists of a pair of diagrams. The upper diagram of each pair illustrates the coded instruction, and the lower illustrates the assembled Instruction Word.

A brief explanation of the diagrams follows. (a) shows the instructions with a straight-forward actual operational address. The decimal address is converted and assembled directly into the Address field of the Instruction Word. (b) indicates the address adjustment. The Assembly Program will have previously assigned an address to SWITCH (in this example, 1454_8). This previously assigned address is then adjusted and assembled in the Address field of the Instruction Word in binary. (c) gives an example of address modification. Again, the assembly program will have previously assigned an address to LABEL (2377_8). The binary equivalent of this address is assembled in the Address field of the Instruction Word. The binary equivalents of M and L are also assembled in their respective M and L fields of the Instruction Word. Modification will occur before the execution of the instruction during the stored program's operations. (FIGURE 7 is located on the following page.)

4. THE CHARACTERISTICS OF THE INSTRUCTION TYPES

CH - These instructions manipulate two data locations; one is in Process Storage specified by W_e or by W*; the other one is in a register specified by R and S. However, there are certain restrictions, as follows: the use of the Channel Service Register as R register is

LINE	NAME		OP CODE		R	S		F		ADDRESS, ML
3 4 5	6 7 8 9 10 11	12	13 14 15 16	17	18	19 20	21	22	23	24
			U N L		Y	9				1376

a)

ADDRESS			R	S		M	L	F	P	OP-CODE	P
47 32			29 27	26 23		20 18	17 14	13 12	11	10 1	
0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0		0	0 1 0	1 0 0 1	0	0 0 0 0	0 0 0 0	0 0	0	1 0 1 0 0 0 0 0 1 0	1

LINE	NAME		OP CODE		R	S		F		ADDRESS, ML
3 4 5	6 7 8 9 10 11	12	13 14 15 16	17	18	19 20	21	22	23	24
			I O R		M	5				SWITCH+2

b)

ADDRESS			R	S		M	L	F	P	OP-CODE	P
47 32			29 27	26 23		20 18	17 14	13 12	11	10 1	
0 0 0 0 0 0 1 1 0 0 1 0 1 1 1 0		0	0 1 1	1 0 1 0	0	0 0 0 0	0 0 0 0	0 0	0	1 0 0 1 0 0 0 0 1 0	1

LINE	NAME		OP CODE		R	S		F		ADDRESS, ML
3 4 5	6 7 8 9 10 11	12	13 14 15 16	17	18	19 20	21	22	23	24
			L O D		Z	7				LABEL, X5

c)

ADDRESS			R	S		M	L	F	P	OP-CODE	P
47 32			29 27	26 23		20 18	17 14	13 12	11	10 1	
0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1		0	0 1 1	1 0 1 1	0	0 0 0 1	0 1 0 1	0 0	1	1 0 1 1 0 0 0 0 1 0	1

FIGURE 7: Assembled Instruction Words

limited as shown in the proper Instruction Tables; all exclusive OR instructions must specify Register X as R Register. AND instructions cannot specify the Interface Data Register as R Register.

AL - These instructions manipulate two locations; one is the Operational Address Register; the other one is an Address or Limit in a Process Storage Word.

SS - These instructions manipulate a Control Storage and a Process Storage location, acting on either the entire word or on the contents of the Address field in both locations. The Process Storage location is specified by W_e or by W*, and the Control Storage location is given as a seven bit address in the RS combined field (bit positions 29-23) of the Instruction Word. For SS instructions executed in the Channel Service Mode, the machine logic uses the contents of the seven low-order bits in the Channel Service Register to address the Control Storage location.

CT - Only one of the seven Branch instructions, Branch Indirect, BRA* (ID No. 70) can be used with address modification by ML. The Branch On Test; BRT, W, MMASK (ID No. 75) is the only instruction which refers to M as a data-manipulating register and not as a modification register. The reason is that the R and S fields are part of the area in the Instruction Word which contains the mask. On the coding line, M must be followed immediately by the bit configuration of the eleven bit mask expressed in four octal digits. The highest number which may be used as a Mask is 3777_8 .

5. THE INSTRUCTION SET

DESCRIPTIVE INDEX

(Descriptive Index continued)

	Identification Numbers	Reference to Tables
3.) EXCLUSIVE OR INSTRUCTIONS Exclusive Or character bits with bits of X Register into X Register.	15 - 22	Table C
4.) INCLUSIVE OR INSTRUCTIONS Inclusive Or character bits with bits of R Register into R Register.	23 - 30	Table D
5.) OR TO PROCESS STORAGE INSTRUCTIONS Inclusive Or character bits with bits of R Register into character field in PS.	31 - 36	Table E
6.) AND INSTRUCTIONS AND character bits with bits of R Register into R Register.	37 - 44	Table F
TYPE II - AL - ADDRESS AND LIMIT MOVING INSTRUCTIONS		
1.) GET INSTRUCTIONS Move Address or Limit, or inclusive OR Address or Limit (all 16 bits) from a PS word to OAR.	45 - 52	Table G
2.) PUT INSTRUCTIONS Move the contents of OAR into Address or Limit of PS word.	53 - 56	Table H
TYPE III - SS - STORAGE TO STORAGE DATA TRANSFER INSTRUCTIONS		
1.) TRANSMIT ADDRESS INSTRUCTIONS Transfer Address from one word to another between Storages, or inclusive OR Address (all 16 bits) from a PS word into Address of a CS word.	57 - 62	Table I

(Descriptive Index continued)

	Identification Numbers	Reference to Tables
2.) MOVE WORD INSTRUCTIONS Transfer entire word from one Storage to another, or inclusive OR an entire PS word (all 47 bits) into a CS word.	63 - 68	Table J
TYPE IV - CT - CONTROL INSTRUCTIONS		
1.) BRANCH INSTRUCTIONS Unconditional and conditional Branch Instructions.	69 - 75	Tables K & L
2.) COMPARE ADDRESS TO LIMIT INSTRUCTIONS Compare Address and Limit of a PS word and skip if unequal.	76 - 77	Table M

TABLE A
LOAD INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		ENTIRE CONTENTS OF REGISTER R ARE CLEARED:	S LOW-ORDER BITS MOVED INTO LOW-ORDER PORTION OF R FROM CHARACTER ADDRESSED BY:	ONE'S COMPLEMENT OF S LOW-ORDER BITS MOVED INTO R FROM CHARACTER ADDRESSED BY:	16 BIT ADDRESS (W*) INCREMENTED BY ONE AT LOCATION:	INCREMENTED ADDRESS (W*) TESTED FOR FIVE LOW ORDER ONES. IF NOT ONES, SKIP:	CHARACTER UNCHANGED AT PS LOCATION:	NUMBER OF CYCLES
		OCTAL	MNEMONIC							
1.	LOAD CHARACTER	1302	LOD	Yes	W_e				W_e	1
2.	LOAD CHARACTER INDIRECT	1312	LOD*	Yes	W*				W*	2
3.	LOAD CHARACTER AND INCREMENT	1322	LOI	Yes	W_e		W_e		W_e	1
4.	LOAD CHARACTER INDIRECT AND INCREMENT	1332	LOI*	Yes	W*		W_e	Yes	W*	2
5.	LOAD COMPLEMENTED CHARACTER	1342	LDC	Yes		W_e			W_e	1
6.	LOAD COMPLEMENTED CHARACTER INDIRECT	1352	LDC*	Yes		W*			W*	2
7.	LOAD COMPLEMENTED CHARACTER AND INCREMENT	1362	LCI	Yes		W_e	W_e		We	1
8.	LOAD COMPLEMENTED CHARACTER INDIRECT AND INCREMENT	1372	LCI*	Yes		W*	W_e	Yes	W*	2

NOTE: A LOAD instruction naming the Channel Service Register as register R will reset only the 4 high-order bit positions (bits 11 - 8) of the CSR.

TABLE B
UNLOAD INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		CHARACTER CLEARED AT PS LOCATION:	S LOW ORDER BITS OF REGISTER R MOVED INTO LOW-ORDER PORTION OF CHARACTER ADDRESSED BY:	ONE'S COMPLEMENT OF S LOW-ORDER BITS OF R MOVED INTO LOW-ORDER PORTION OF CHARACTER ADDRESSED BY:	16 BIT ADDRESS (W*) INCREMENTED BY ONE AT LOCATION:	INCREMENTED ADDRESS (W*) TESTED FOR FIVE LOW-ORDER ONES. IF NOT, ONES, SKIP:	CONTENTS OF REGISTER R ARE UNCHANGED	NUMBER OF CYCLES
		OCTAL	MNEMONIC							
9.	UNLOAD CHARACTER	1202	UNL	W_e	W_e				Yes	1
10.	UNLOAD CHARACTER INDIRECT	1212	UNL*	W*	W*				Yes	2
11.	UNLOAD CHARACTER INDIRECT AND INCREMENT	1232	ULI*	W*	W*		W_e	Yes	Yes	2
12.	UNLOAD COMPLEMENTED CHARACTER	1242	ULC	W_e		W_e			Yes	1
13.	UNLOAD COMPLEMENTED CHARACTER INDIRECT	1252	ULC*	W*		W*			Yes	2
14.	UNLOAD COMPLEMENTED CHARACTER INDIRECT AND INCREMENT	1272	UCI*	W*		W*	W_e	Yes	Yes	2

TABLE C
EXCLUSIVE OR INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		S LOW ORDER BITS OF X ARE EXCLUSIVE OR'D INTO X WITH S CORRESPONDING LOW-ORDER BITS OF CHARACTER ADDRESSED BY:	S LOW ORDER BITS OF X ARE EXCLUSIVE OR'D INTO X WITH ONE'S COMPLEMENT OF THE S CORRESPONDING LOW-ORDER BITS OF CHARACTER ADDRESSED BY:	11 MINUS S HIGH ORDER BITS OF REGISTER X UNCHANGED:	16 BIT ADDRESS (W*) INCREMENTED BY ONE AT LOCATION:	INCREMENTED ADDRESS (W*) TESTED FOR FIVE LOW ORDER ONES. IF NOT ONES, SKIP:	CHARACTER UNCHANGED AT PS LOCATION:	NUMBER OF CYCLES
		OCTAL	MNEMONIC							
15.	EXCLUSIVE OR	1101	XOR	W_e		Yes			W_e	1
16.	EXCLUSIVE OR INDIRECT	1111	XOR*	W*		Yes			W*	2
17.	EXCLUSIVE OR AND INCREMENT	1121	XOI	W_e		Yes	W_e		W_e	1
18.	EXCLUSIVE OR INDIRECT AND INCREMENT	1131	XOI*	W*		Yes	W_e	Yes	W*	2
19.	EXCLUSIVE OR COMPLEMENTED	1141	XOC		W_e	Yes			W_e	1
20.	EXCLUSIVE OR COMPLEMENTED INDIRECT	1151	XOC*		W*	Yes			W*	2
21.	EXCLUSIVE OR COMPLEMENTED AND INCREMENT	1161	XCI		W_e	Yes	W_e		W_e	1
22.	EXCLUSIVE OR COMPLEMENTED INDIRECT AND INCREMENT	1171	XCI*		W*	Yes	W_e	Yes	W*	2

NOTE: Exclusive OR Instructions can be used only with register X.

TABLE D
INCLUSIVE OR INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		S LOW-ORDER BITS OF R ARE INCLUSIVE ORD'd INTO R WITH S CORRESPONDING LOW-ORDER BITS OF CHARACTER ADDRESSED BY:	S LOW-ORDER BITS OF R ARE INCLUSIVE OR'D INTO R WITH ONE'S COMPLEMENT OF THE S CORRESPONDING LOW-ORDER BITS OF CHARACTER ADDRESSED BY:	11 MINUS S HIGH-ORDER BITS OF REGISTER R UNCHANGED:	16 BIT ADDRESS (W*) INCREMENTED BY ONE AT LOCATION:	INCREMENTED ADDRESS (W*) TESTED FOR FIVE LOW-ORDER ONES. IF NOT ONES, SKIP:	CHARACTER UNCHANGED AT PS LOCATION	NUMBER OF CYCLES
		OCTAL	MNEMONIC							
23.	INCLUSIVE OR	1102	IOR	W_e		Yes			W_e	1
24.	INCLUSIVE OR INDIRECT	1112	IOR*	W*		Yes			W*	2
25.	INCLUSIVE OR AND INCREMENT	1122	IOI	W_e		Yes	W_e		W_e	1
26.	INCLUSIVE OR INDIRECT AND INCREMENT	1132	IOI*	W*		Yes	W_e	Yes	W*	2
27.	INCLUSIVE OR COMPLEMENTED	1142	IOC		W_e	Yes			W_e	1
28.	INCLUSIVE OR COMPLEMENTED INDIRECT	1152	IOC*		W*	Yes			W*	2
29.	INCLUSIVE OR COMPLEMENTED AND INCREMENT	1162	ICI		W_e	Yes	W_e		W_e	1
30.	INCLUSIVE OR COMPLEMENTED INDIRECT AND INCREMENT	1172	ICI*		W*	Yes	W_e	Yes	W*	2

NOTE: The Channel Service Register cannot be used with Inclusive OR Instructions.

TABLE E
OR TO PROCESS STORAGE INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		S LOW-ORDER BITS OF R ARE INCLUSIVE OR'D INTO S CORRESPONDING LOW-ORDER BITS OF CHARACTER ADDRESSED BY:	ONE'S COMPLEMENT OF S LOW-ORDER BITS OF R ARE INCLUSIVE OR'D INTO S CORRESPONDING LOW-ORDER BITS OF CHARACTER ADDRESSED BY:	11 MINUS S HIGH-ORDER BITS OF CHARACTER UNCHANGED AT PS LOCATION	16 BIT ADDRESS (W*) INCREMENTED BY ONE AT LOCATION:	INCREMENTED ADDRESS (W*) TESTED FOR FIVE LOW-ORDER ONES. IF NOT, SKIP:	ENTIRE CONTENTS OF REGISTER R UNCHANGED	NUMBER OF CYCLES
		OCTAL	MNEMONIC							
31.	OR TO PROCESS STORAGE	1002	ORP	W_e		W_e			Yes	1
32.	OR TO PROCESS STORAGE INDIRECT	1012	ORP*	W*		W*			Yes	2
33.	OR TO PROCESS STORAGE INDIRECT AND INCREMENT	1032	ORI*	W*		W*	W_e	Yes	Yes	2
34.	OR COMPLEMENT TO PROCESS STORAGE	1042	OCP		W_e	W_e			Yes	1
35.	OR COMPLEMENT TO PROCESS STORAGE INDIRECT	1052	OCP*		W*	W*			Yes	2
36.	OR COMPLEMENT TO PROCESS STORAGE INDIRECT AND INCREMENT	1072	OCI*		W*	W*	W_e	Yes	Yes	2

TABLE F
AND INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		S LOW-ORDER BITS OF R ARE AND'D INTO R WITH S CORRESPONDING LOW-ORDER BITS OF CHARACTER ADDRESSED BY:	S LOW-ORDER BITS OF R ARE AND'D INTO R WITH ONE'S COMPLEMENT OF S CORRESPONDING LOW-ORDER BITS OF CHARACTER ADDRESSED BY:	11 MINUS S HIGH-ORDER BITS OF REGISTER R UNCHANGED:	16 BIT ADDRESS (W*) INCREMENTED BY ONE AT LOCATION:	INCREMENTED ADDRESS (W*) TESTED FOR FIVE LOW-ORDER ONES. IF NOT ONES, SKIP:	CHARACTER UNCHANGED AT PS LOCATION:	NUMBER OF CYCLES
		OCTAL	MNEMONIC							
37.	AND	1145	AND	W_e		Yes			W_e	1
38.	AND INDIRECT	1155	AND*	W*		Yes			W*	2
39.	AND AND INCREMENT	1165	ANI	W_e		Yes	W_e		W_e	1
40.	AND INDIRECT AND INCREMENT	1175	ANI*	W*		Yes	W_e	Yes	W*	2
41.	AND COMPLEMENTED	1105	ANC		W_e	Yes			W_e	1
42.	AND COMPLEMENTED INDIRECT	1115	ANC*		W*	Yes			W*	2
43.	AND COMPLEMENTED AND INCREMENT	1125	ACI		W_e	Yes	W_e		W_e	1
44.	AND COMPLEMENTED INDIRECT AND INCREMENT	1135	ACI*		W*	Yes	W_e	Yes	W*	2

NOTE: The Channel Service Register and the Interface Data Register cannot be used with AND instructions.

TABLE G
GET INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		THE CONTENTS OF THE OAR IS REPLACED BY THE ADDRESS IN THE WORD ADDRESSED BY:	THE CONTENTS OF THE OAR IS REPLACED BY THE LIMIT IN THE WORD ADDRESSED BY:	THE CONTENTS OF THE OAR IS INCLUSIVE OR'D INTO THE OAR, WITH THE CORRESPONDING BITS OF THE ADDRESS IN THE WORD ADDRESSED BY:	THE CONTENTS OF THE OAR IS INCLUSIVE OR'D INTO THE OAR WITH THE CORRESPONDING BITS OF THE LIMIT IN THE WORD ADDRESSED BY:	WORD UNCHANGED AT PS LOCATION:	NUMBER OF CYCLES
		OCTAL	MNEMONIC						
45.	GET ADDRESS	0702	GTA	W_e				W_e	1
46.	GET ADDRESS INDIRECT	0712	GTA*	W*				W*	2
47.	GET LIMIT	0302	GTL		W_e			W_e	1
48.	GET LIMIT INDIRECT	0312	GTL*		W*			W*	2
49.	GET AND OR ADDRESS	0502	GOA			W_e		W_e	1
50.	GET AND OR ADDRESS INDIRECT	0512	GOA*			W*		W*	2
51.	GET AND OR LIMIT	0102	GOL				W_e	W_e	1
52.	GET AND OR LIMIT INDIRECT	0112	GOL*				W*	W*	2

TABLE H
PUT INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		THE CONTENTS OF THE OAR REPLACES THE ADDRESS IN THE WORD ADDRESSED BY:	THE CONTENTS OF THE OAR REPLACES THE LIMIT IN THE WORD ADDRESSED BY:	CONTENTS OF OAR IS UNCHANGED:	NUMBER OF CYCLES
		OCTAL	MNEMONIC				
53.	PUT ADDRESS	0602	PTA	W_e		Yes	1
54.	PUT ADDRESS INDIRECT	0612	PTA*	W*		Yes	2
55.	PUT LIMIT	0202	PTL		W_e	Yes	1
56.	PUT LIMIT INDIRECT	0212	PTL*		W*	Yes	2

TABLE I
TRANSMIT ADDRESS INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		THE ADDRESS IN THE CS WORD REPLACES THE ADDRESS IN THE PS WORD ADDRESSED BY:	WORD UNCHANGED AT CS LOCATION:	THE ADDRESS IN THE CS WORD IS REPLACED BY THE ADDRESS IN THE PS WORD ADDRESSED BY:	THE ADDRESS IN THE CS WORD IS INCLUSIVE OR'D INTO THE CS WORD WITH THE ADDRESS IN THE PS WORD ADDRESSED BY:	WORD UNCHANGED AT PS LOCATION:	NUMBER OF CYCLES
		OCTAL	MNEMONIC						
57.	TRANSMIT ADDRESS TO PROCESS STORAGE	0606	TAP	W_e	Yes				2
58.	TRANSMIT ADDRESS TO PROCESS STORAGE INDIRECT	0616	TAP*	W*	Yes				2
59.	TRANSMIT ADDRESS TO CONTROL STORAGE	0706	TAC			W_e		W_e	2
60.	TRANSMIT ADDRESS TO CONTROL STORAGE INDIRECT	0716	TAC*			W*		W*	2
61.	TRANSMIT AND OR ADDRESS TO CONTROL STORAGE	0506	TOC				W_e	W_e	2
62	TRANSMIT AND OR ADDRESS TO CONTROL STORAGE INDIRECT	0516	TOC*				W*	W*	2

NOTE: The Control Storage address must be specified in the R-S combined field with the exception of instructions in the Channel Service Mode Program. In this Mode, the CS address is automatically taken from the Channel Service Register.

TABLE J
MOVE WORD INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		THE CONTENTS OF THE CS WORD REPLACES THE CONTENTS OF THE PS WORD ADDRESSED BY:	WORD UNCHANGED AT CS LOCATION:	THE CONTENTS OF THE CS WORD IS REPLACED BY THE CONTENTS OF THE PS WORD ADDRESSED BY:	THE CONTENTS OF THE CS WORD IS INCLUSIVE OR'D INTO THE CS WORD WITH THE CONTENTS OF THE PS WORD ADDRESSED BY:	WORD UNCHANGED AT PS LOCATION:	NUMBER OF CYCLES
		OCTAL	MNEMONIC						
63.	MOVE WORD TO PROCESS STORAGE	0206	MWP	W_e	Yes				2
64.	MOVE WORD TO PROCESS STORAGE INDIRECT	0216	MWP*	W*	Yes				2
65.	MOVE WORD TO CONTROL STORAGE	0306	MWC			W_e		W_e	2
66.	MOVE WORD TO CONTROL STORAGE INDIRECT	0316	MWC*			W*		W*	2
67.	MOVE WORD AND OR TO CONTROL STORAGE	0106	MOC				W_e	W_e	2
68.	MOVE WORD AND OR TO CONTROL STORAGE INDIRECT	0116	MOC*				W*	W*	2

NOTE: The Control Storage address must be specified in the R-S combined field with the exception of instructions in the Channel Service Mode Program. In this Mode, the CS address is automatically taken from the Channel Service Register.

TABLE K
BRANCH INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		CONDITION TO BE MET FOR TRANSFER: S LOW-ORDER BITS OF REGISTER R ARE:	LOCATION OF NEXT INSTRUCTION		MODIFICATION OF W BY ML IS PERMITTED:	R IS UNCHANGED:	NUMBER OF CYCLES
		OCTAL	MNEMONIC		IF CONDITION IS MET	IF CONDITION IS NOT MET			
69.	BRANCH	C707	BRA	IRRELEVANT	W (UNCONDITIONAL TRANSFER)		No		1
70.	BRANCH INDIRECT	C717	BRA*	IRRELEVANT	W* (UNCONDITIONAL TRANSFER)		Yes		1
71.	BRANCH ON ZERO	C704	BRZ	ALL ZEROS	W	NEXT INSTRUCTION IN SEQUENCE	No	Yes	1
72.	BRANCH ON ZERO INDIRECT	C714	BRZ*	ALL ZEROS	W*	NEXT INSTRUCTION IN SEQUENCE	No	Yes	1
73.	BRANCH ON ONES	C744	BRO	ALL ONES	W	NEXT INSTRUCTION IN SEQUENCE	No	Yes	1
74.	BRANCH ON ONES INDIRECT	C754	BRO*	ALL ONES	W*	NEXT INSTRUCTION IN SEQUENCE	No	Yes	1

TABLE L
BRANCH ON TEST INSTRUCTION

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		CONTENTS OF POSITIONS 33 THROUGH 23 OF THE INSTRUCTION WORD IS INCLUSIVE OR'D WITH THE CONTENTS OF REGISTER M.		MODIFICATION OF INSTRUCTION ML IS PERMITTED:	INDIRECT ADDRESSING IS PERMITTED:	CONTENTS OF INSTRUCTION WORD AND REGISTER M ARE UNCHANGED:	NUMBER OF CYCLES
		OCTAL	MNEMONIC	IF RESULT IS ALL ONES	IF RESULT IS NOT ALL ONES				
75.	BRANCH ON TEST	0544	BRT	GO TO INSTRUCTION ADDRESSED BY THE ADDRESS (BITS 47-34) OF THE BRT INSTRUCTION WORD	GO TO NEXT INSTRUCTION IN SEQUENCE	No	No	Yes	1

Note: On the coding line, the format: BRT ADDRESS, MXXXX is used, where M is the register and XXXX is the four-digit octal representation of the MASK to be stored in bit positions 33 - 23 of the Instruction Word. The MASK must always consist of 4 digits.

TABLE M

COMPARE ADDRESS TO LIMIT INSTRUCTIONS

INSTRUCTION ID NUMBER	DESCRIPTION	OP CODE		COMPARE ADDRESS TO LIMIT OF THE SAME WORD ADDRESSED BY W_e:		ADDRESS IN THE WORD AT W_e IS INCREMENTED BY ONE AFTER THE COMPARISON:	NUMBER OF CYCLES
		OCTAL	MNEMONIC	IF IDENTICAL	IF NOT IDENTICAL		
76.	COMPARE ADDRESS TO LIMIT	0003	CAL	GO TO NEXT INSTRUCTION IN SEQUENCE	SKIP NEXT AND GO TO INSTRUCTION AFTER NEXT	No	1
77.	COMPARE ADDRESS TO LIMIT AND INCREMENT	0023	CAI	GO TO NEXT INSTRUCTION IN SEQUENCE	SKIP NEXT AND GO TO INSTRUCTION AFTER NEXT	Yes	1

SECTION XI

MISCELLANEOUS ASPECTS

1. ADDRESSING SYSTEMS

A. CONTROL STORAGE

Control Storage operates with a 7 bit binary address system. Addresses range from 000_{10} to 127_{10} (i. e., from 000_8 to 177_8). The addresses of the six Process Words and the two Scratch Words are as follows:

CS WORD	ADDRESS IN DECIMAL	ADDRESS IN OCTAL
SCRATCH WORD	30	36
SERVICE MODE PWD	31	37
SCRATCH WORD	62	76
CHANNEL SERVICE MODE PWD	63	77
OUT MODE PWD	94	136
COPY MODE PWD	95	137
IN MODE PWD	126	176
NORMAL MODE PWD	127	177

The remaining locations will be assigned to Channel Words and to Error Channel Words (the latter only if there are high-speed Communication Channels on a system), as required by the channel configuration of each system.

Whenever a CS location is affected by an instruction (i. e., SS instruction), the 7 bit binary address is located in bit positions 29-23 (i. e., R and S as a combined field) of the Instruction Word. On the Coding sheet, the CS address must be given as an actual address in decimal in columns 18-20 (i. e., R and S combined columns); it may be expressed by a one, two, or three digit decimal number, the low-order digit written in column 20.

Whenever the IBM 7750 operates in the Channel Service Mode, the address located in the seven low-order bit positions of the Channel Service Register is used by the machine logic to access the Control Storage Word for the execution of an SS instruction. Therefore, the Control Storage address need not to be given in SS instructions implemented in the Channel Service Mode Program.

B. PROCESS STORAGE

Process Storage operates with a 16 bit binary address system. Addresses range as follows:

Size of Machine In Words	Character Addresses In Decimal	Character Addresses In Octal
4K	0 - 16383	0 - 037777
8K	0 - 32767	0 - 077777
16K	0 - 65535	0 - 177777

Since Process Storage is divided into blocks of eight words and the eleven high-order bits of the 16 bit addresses are the same for all 8 words and, consequently, for all 32 characters within a block; each 16 bit address may be interpreted as a three-part address as follows:

Bits 16 - 6 (i.e., the eleven high-order bits) represent a block address.

Bits 5 - 3 represent a word address within each block.
These three bits range from 000_2 to 111_2 to indicate the first (0_8) to the eighth (7_8) word of each block respectively.

Bits 2 - 1 (i.e., the two low-order bits) represent the character address within each word.
They range from 00_2 to 11_2 to refer to A, B, C, and D characters, respectively.

It is obvious that the 16 bit address of the first character of each block (i.e., the "A" character of the first word of each block) ends in five Zeros, and the address of the last character (i.e., the "D" character of the eighth word of each block, or the Block Control Character) ends in five Ones.

Process Storage is organized and operated on a character address basis. Whenever an actual address is intended, the decimal address of the character must be written in the variable field of the coding line beginning with column 24. For a word address, the first character address ("A") of the word should be given in decimal. If the Assembly Program assigned a word location to a label (Name) upon a command given by a Control (Pseudo) Statement, this label refers to the word, or to the first character ("A") of the word, depending on the type of instruction with which it is used. In this case, an address adjustment

of +1, +2, +3 following the label (Name) refers to the "B", "C", and "D" characters, respectively, in the word in which the "A" character is represented by the label (Name) itself. A +4 adjustment refers to the "A" character of the next consecutive word, and so on. A -1 adjustment refers to the "D" character of the preceding word, and so on. A Branch instruction given with an "*+8" address will cause a branch to the second instruction in sequence, and with an "*-12" address will cause a branch to the third instruction preceding the one with which it is used. If the instruction is other than a CH instruction, it will be executed on the contents of the same Process Storage Word whether the 16 bit Address (bits 47-32) in the IWD contains 00, 01, 10, or 11 in the two low-order bit positions because the two low-order bits are ignored by the machine logic when an instruction refers to a word and not to a character. Everything written into the variable field of the coding line beginning with Column 24 up to the first blank or comma, whichever comes first, will be interpreted or computed by the Assembly Program as a Process Storage address and will be placed into the Address field (bit position 47-32) of the Instruction Word as "W". After the first blank, everything will be considered as a comment. After the first comma, if no blank appears before it in any position beginning with column 24, M must be named by its identifying letter as modification register followed by L, the number of low-order bits to be used from Register M as modifier.

C. ADDRESS MODIFICATION

Address modification by ML results in a 16 bit Process Storage address composed of 16 minus L high-order bits of W, and L low-order bits from Register M. Before the execution of an instruction, if modification is not required, W is moved from PSAR2 into PSAR1 and becomes W_e. In this case, $W_e = W$. If modification is required, L number of low-order bits of M are moved into the low-order bit positions of PSAR1 and 16 minus L high-order bits are moved from PSAR2 into the high-order bit positions of PSAR1. In this case, W_e is the modified version of W. In both cases, W_e refers to a data location (a character or a word, depending on the type of the instruction) if indirect addressing is not used, and to a Process Storage word, the Address field (bits 47-32) of which contains the data address (W*), if indirect addressing is used. The indirect address (W*) will be interpreted by the machine logic as a character address or as a word address according to the type of instruction with which it is used.

2. FLEXIBLE MEMORY UTILIZATION: BLOCKS, CHAINS, QUEUES.

The unique organization of Process Storage provides unlimited flexibility in memory usage. This is one of the most significant features of the IBM 7750

since the size of the information units, commonly called messages, flowing through a "TELE-PROCESSING" System is unpredictable. Each application may have different characteristics and requirements; and, in most applications, messages of variable lengths will be transmitted. Therefore, the IBM 7750 must be prepared to handle messages of different sizes. This alone would not constitute a problem if only a few memory areas were to be used for storing and holding input, output, or other unprocessed or processed data as happens in other than communication-based data processing systems; each reserved area would simply be sized to the maximum length of the data-unit to be stored in that area. In a "TELE-PROCESSING" System, however, the number of such assigned areas may run high since each Communication Channel must have its own areas for input or output (or, in case of a half-duplex channel, for both) and for processed data. A further characteristic of a "TELE-PROCESSING" system is the constant data flow from and to the Communication Channels, which may make it necessary to maintain adequate buffer storage areas for more than one information unit per function (i.e., input, output, process areas) for each Communication Channel. Efficient memory area assignment, therefore, would not be possible on a permanent allocation basis; a large portion of the core memory would be tied down constantly, resulting in a low factor of utilization.

A. BLOCK - The entire Process Storage is divided into blocks. A block consists of eight consecutive words. As explained in Chapter 1 of this section, each 16 bit binary Process Storage character address may be interpreted as a three-part address: the 11 high-order bits represent a block address, the next 3 bits represent a word within a block, varying from 000_2 to 111_2; the 2 low-order bits represent a character within a word varying from 00_2 to 11_2. It is evident that the first character address of each block has a 16 bit binary address ending in five Zeros, and the last (32nd) character address of each block has a 16 bit binary address ending in five Ones.

B. CHAIN - Whenever a block is connected to another in order to form a larger memory area for storing coherent data, a Chain is composed. It may be of any size, two blocks or more, randomly linked together. A chain is not necessarily composed of Data Word blocks; blocks containing Limit Words may also be chained together. Closed chains may be formed by placing the address of the first block into the BCC of the last block.

C. BLOCK CONTROL CHARACTER - The last (32nd) character of each block (i.e., the "D" character in the eighth word) having five Ones in the low-order bit positions of its 16 bit PS address is called the Block Control Character (BCC). For blocks linked together in a chain, this character position is used to record the block address of the next block (which may be located anywhere in Process Storage) connected to the block containing the Block Control Character. Since the address of the first

character in every block ends in five Zeros, and the 11 bit block address is stored in the Block Control Character, the address of the first character of the next block in the chain can be easily constructed by adding five trailing Zeros to the 11 bit block address. Blocks, and parts of blocks (words) assigned for holding permanent information such as tables, constants, etc., do not have to be linked together; therefore, in such applications, the Block Control Character may be used for holding data. (See Appendix IV - Chart 5 for Block Chaining.)

D. AVAILABLE MEMORY SPACE INVENTORY Normally, the unused blocks are kept in one long chain throughout the entire operation of the IBM 7750. At the beginning of the operation, all blocks not having assignments will be chained together by the program (or a utility program). Each time a new block is required, the first of the available blocks will be removed from the beginning of the chain. Whenever a block is no longer needed, it will be returned and linked to the end of this chain. These functions must be taken care of by the program. By this flexible method, no significant portion of memory will remain unused, because a new block will be assigned from the chain of available blocks when it is needed and a block will be returned to the chain as soon as it is no longer required. One single Limit Word can control the available memory space at all times. The Address field of this Limit Word contains the first character address of the first available block, and the Limit field contains the last character address (BCC) of the last available block. The D character field is used to record the block count. When a block is removed from the chain, the block count is decremented; when returned, it is incremented by the program. The block count must be checked by the program periodically in order to maintain a safe amount of buffer space; if it falls below the program-determined "safety minimum", proper program routines must change certain operations (e.g.,increase the output, decrease the input operation) until the available buffer space is properly increased. (See Appendix IV - Chart 6 for Available Memory Space Inventory.) Examples are shown below for the coding of a new block assignment (FIGURE 9) and for returning a block to the available memory space inventory (FIGURE 11).

Example for a new block assignment (FIGURES 8 and 9):

Input characters are processed from the input area of a channel, and the processed characters are stored in a memory area controlled by LWD1. An End of Block condition has been detected in LWD1 and a new block assignment is necessary for storing the next group of characters.

Labels (Names) used in the routine are:

AMSLWD	Limit Word controlling available memory space (See Appendix IV - Chart 6).
LWD1	Limit Word controlling process area. Its contents at the time the routine is to be executed are shown in FIGURE 8.

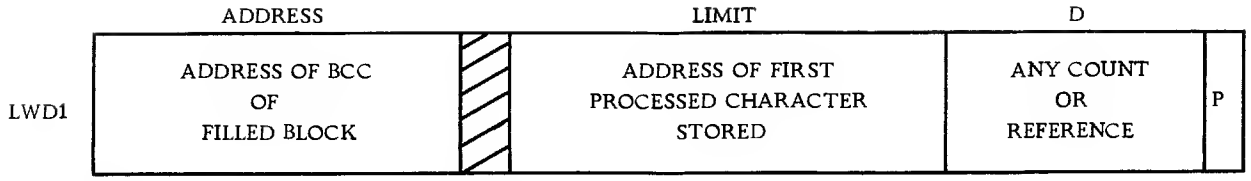


FIGURE 8: Process Area Controlling Limit Word

NAME	OP CODE	R	S	F	ADDRESS, ML	COMMENTS
	GTA			2	AMSLWD	GET ADDR. FROM AMSLWD
	UNL*	Y	1	2	LWD1	PLACE INTO BCC OF PRESENT BLOCK.
	PTA			2	LWD1	PUT INTO ADDR. OF LWD1
	UNL	Y	1	2	NEXT	PLACE INTO ADDR. OF NEXT INSTR.
NEXT	LOD	Y	1	2	31	BCC OF FIRST AVAIL. BLOCK.
	UNL	Y	1	2	AMSLWD	BCC TO ADDR. AMSLWD.
	LOD	Z	1	2	AMSLWD+3	BLOCK COUNT TO Z REGISTER.
	UNL	Z	1	1	AMSLWD+3	DECREMENT COUNT AND STORE IN D.

FIGURE 9: Coding Example for Removing a Block from Available Memory Space.

Explanation: For the program shown in FIGURE 9, the first available block must be removed from the available chain and linked to the already filled block. The first character address of the newly assigned block must be placed in the Address field of LWD1, and the address of the removed block must be replaced with the address of the next available block in the Address field of AMSLWD. The block count must be updated.

Example for returning a block to available memory space inventory (FIGURES 10 and 11):
Characters are obtained from an input area. After processing, they are stored in another area controlled by another Limit Word. Therefore, when all the characters of a block have been processed, the block is no longer needed and should be returned to the available memory space inventory.

Labels (Names) used in the routine are:

AMSLWD	Limit word controlling available memory space.
IPLWD	Limit word controlling the input area. Its contents are shown in FIGURE 10.

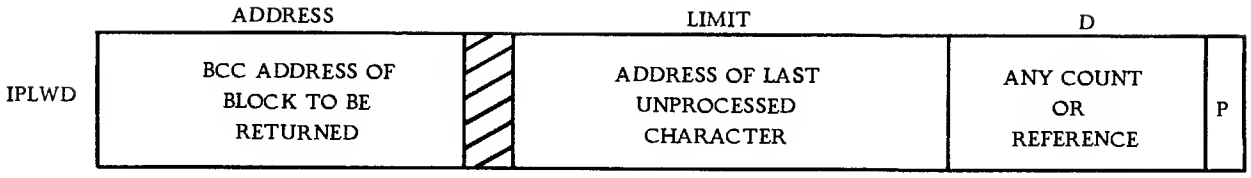


FIGURE 10: Input Area Controlling Limit Word

NAME	OP CODE		R	S	F	ADDRESS, ML	COMMENTS
	LOD	*	Y	1,1	2	IPLWD	BCC TO Y REGISTER
	LOI		Z	1,1	2	IPLWD	BLK. ADDR. TO Z. CHANGE 5 ONES TO ZEROS
	UNL		Y	1,1	2	IPLWD	NEXT BLOCK ADDR. TO IPLWD
	GTL				2	AMSLWD	BCC ADDR. OF LAST BLOCK TO OAR
	PTA				2	NEXT	BCC ADDR. OF LAST BLOCK TO INSTRCTN.
NEXT	UNL		Z	1,1	2		CHAIN RELEASED BLK. TO PREVIOUS LAST
	UNL		Z	1,1	2	AMSLWD+2	BCC ADDR. OF NEW LAST BLOCK TO AMSLWD
	LDC		Z	1,1	2	AMSLWD+3	BLOCK COUNT TO Z REGISTER
	ULC		Z	1,1	1	AMSLWD+3	STORE INCREMENTED COUNT BACK TO D

FIGURE 11: Coding Example for Returning a Block to Available Memory Space.

Explanation: For the program shown in FIGURE 11, the address of the next input block must be obtained from the BCC and must be placed into the Address field of the IPLWD with 5 trailing zeros. The released block must be chained to the current last available block. The BCC address of the released block must be placed in the Limit field of AMSLWD. The block count must be updated.

E. END OF BLOCK CONDITION When characters are assembled in, or transmitted from, the Assembly Area of a Channel Word, after a character is stored in, or obtained from, Process Storage, the Address in the Channel Word is automatically incremented by one. After the increment, the five low-order bits of the Address are tested for all Ones. The same automatic function takes place in the Address field of the Copy Word after the transmission of each character to or from the Processor. Limit Words control the data holding chains in Process Storage. After the execution of "indirect and increment" instructions (i. e., each CH instruction having an I as

the third and an * as the fourth symbol in the mnemonic Operation Code), the Address (W*) in the accessed Limit Word is incremented by one and the incremented Address is tested for five low-order Ones.

In all above cases, when the "five Ones Test" is performed, and five Ones are detected in the 5 low-order bit positions of the 16 bit Address, the machine logic recognizes an End Of Block condition. Upon recognition, if the tested Address is located in a Channel Word, an attempt is made to obtain Channel Service: when executed, the Channel Service Mode Program provides the first character address of the next block for the Channel Word. In the Copy Mode, a fully automatic procedure provides the new Address for the Copy Word. When processing in Process Storage, since the referenced instructions are skip-type instructions, the otherwise skipped next instruction in sequence will be executed, branching to a routine in which the new Address for the Limit Word is provided by the program.

F. QUEUE A PS area, containing a certain amount of coherent data is controlled by a Limit Word. In some cases, several such data-complexes must be "lined up" for a following operation (output, processing) which affects the same communication channel or the Processor. This can be done by assigning Limit Words in consecutive PS locations for controlling each area and controlling these Limit Words by a Master Limit Word (Queue Limit Word). The associated data units are called a queue. The individual Limit Words may exceed the capacity of one block in which case the blocks must be chained together. In such a two level Limit Word operation each data unit can be accessed through its Limit Word located in its Limit Word Chain which is controlled by the Queue Limit Word.

3. LIMIT WORDS AND THEIR APPLICATIONS

Process Storage blocks, when linked together in a chain to provide memory areas of sufficient size for constantly changing data, must be controlled in some way by the program. Not only is the data in a memory area being changed from time to time, but also the blocks composing the area will be blocks from different memory locations, and the size of the controlled area may change (i. e., become larger or smaller) as the program proceeds. Therefore, the program cannot use a definite memory location to refer to its data except for tables, constants, and Limit Words.

The Limit Word is the controlling word for a chain or a memory area. Sometimes, a large area of numerous blocks may be controlled by a Limit Word, and the area may, or may not, begin with the first character field of the first block and may, or may not, end in the last character field (i. e., the 31st

character, since the 32nd character is the BCC) of the last block. Sometimes, as data processing progresses,or when areas are being used for certain special purposes predetermined by the program, the area controlled by a Limit Word may even be smaller than a block size. Since the physical location and the size of the areas are constantly changing, but the same Limit Words with known permanent locations will be controlling all areas used for preassigned purposes throughout the entire operation, the program has a convenient way for operating on these areas through their controlling Limit Words.

For processing data, character by character, convenient instructions are available for addressing data characters indirectly through the Address (W*) located in the area controlling Limit Word. The Address may,or may not,be incremented after the instruction has been executed, as required. The automatic End Of Block recognition feature is associated with a number of such instructions by the use of micro-codes. Instructions are available to compare the Address and the Limit in the same Limit Word with,or without,a following Address increment, and to go to the next instruction,or skip it if the compare is equal or unequal, respectively.

There are numerous ways of using Limit Words in a program. The programmer may even find some unusual ways to use them to satisfy his purposes. Even a combination of a Limit Word and a Data Word may be used; for example, the Address field may be used for an address or count,and the C and D fields for some other information not exceeding 11 bits in length. Some instructions may treat such a word as a Limit Word (when operating on the contents of the Address field) and some others,as a Data Word (when operating on the C or D field or, on the 11 high order bits - A field - of the Address). Limit Words may also be used on more than one level in an operation; for example, a string of Limit Words may be located in consecutive Word locations within each block of a chain, and the complex of these words may be controlled by another Limit Word, which may be called "Master Limit Word ".

A few examples are given in FIGURE 12 for the typical uses of the Limit Word format.

4. THE ORGANIZATION OF TABLES. TABLE LOOKUP

Numerous programmed functions of the IBM 7750 "call" for the application of tables, either by necessity or by logical choice, as the most feasible method. Some examples of the most frequently used table lookup operations are: character-validity check, code conversion, editing, shifting, branching to the first instruction of a routine (e. g., in the "area scanning" operation)and branching to a special routine upon the recognition of a special character (e. g., Start of Message, End of Message, Carriage Return, etc.).

PURPOSE	ADDRESS	LIMIT	D FIELD
Input Area controlling LWD for a channel	Address of next character to be processed in Input Area	Address of character position after current last character in Input Area (obtained from CWD)	Any count or block address, or the address of the CWD
Process Area controlling LWD for a channel	Address of next processed (from Input Area) character to be stored	Address of first processed character stored in this area	Any count or block address, or the address of the CWD
Output Area controlling Master (Queue) LWD for a channel (one per channel)	Address of first Output Message LWD in current Output Message LWD string (chain)	Address of last Output Message LWD in current Output Message LWD string (chain)	Count of Output Message LWD's in current string (chain)
Output Message LWD (several per channel)	Address of first character of a complete output message	Any information	Any information (e. g. address of last block of message for purpose of releasing)
Input Area from CPU controlling LWD	Address of first unprocessed character in Input Area from CPU	Address of last unprocessed character + 1 in Input Area from CPU	Any information
Output Area for CPU controlling Master (Queue) LWD (one)	Address of first Image Copy Word in current Image Copy Word string (chain)	Address of last Image Copy Word in current Image Copy Word string (chain)	Count of Image Copy Words in current string (chain)
Image Copy Word (several)	Address of first character of an information unit to be transferred to the CPU	Address of last character + 1 of the same information unit	All Zeros (or another binary number for first time, as required)
Available Memory Space LWD (usually one)	Address of first character of first available block	Address of last character (BCC) of last available block	Block-count (i. e. number of available blocks in empty buffer storage)
Polling Chain LWD (at least one per half duplex channel and per Sending line of a full duplex channel)	Address of next Terminal identifying character to be used for Polling	Address of last Terminal identifier in polling chain or Address of first identifier in original chain if EOB feature is used to return to beginning of closed polling chain	Address of first Terminal identifier in polling chain or all Zeros if EOB feature is used to return to beginning of closed polling chain
Increment Counter	16 bit binary number to be incremented by one in a program loop	16 bit binary number with which the incremented number is to be compared (if any)	Any information

FIGURE 12 : Examples for the Typical Uses of the Limit Word Format

The IBM 7750 has no arithmetic unit because its main functions are to forward the input data to the Processor and to transmit the output data to Terminals, after the necessary code conversion and editing (i. e., deleting or adding functional characters), as quickly as possible and to control the entire operation of the network. The actual data processing, including the required arithmetic operations, can be well and most efficiently taken care of by the Processor; therefore, a request for the IBM 7750 to perform such operations would not be justified. The Processor has quick access to its record-keeping auxiliary units, and in most cases, some of these records must be accessed before the required arithmetic operations can be performed. The IBM 7750 has no access to these records. Simple arithmetic functions, however, such as decrementing, incrementing, and keeping and checking counts can be taken care of by the IBM 7750 just as efficiently as by any other computer. Even the addition and subtraction of smaller numbers can be performed economically by the application of tables.

The logical method for the application of table lookup operations is to use a bit pattern (a binary number assigned to a specific function by the program, or a code bit pattern of an input or a Processor-provided output character) as an indexing factor and, through it, access and obtain the desired entry from a table.

The entries in a table may be characters, addresses or full words. When the table entries are addresses--located in the Address field (bits 47-32) of the table words--the Limit field (bits 27-12) and the D field (bits 11-1), or the C (bits 22-12) and D (bits 11-1) fields (depending on the internal organization of the table words), may be used for holding reference items for the accessed area.

The organization of tables must follow certain basic rules. If the entries in the table are characters, (i. e., each entry is stored in one of the character fields of a word), the table must be located in Process Storage so that the first character (i. e., entry) address of the table (i. e., the 16 bit PS address of the first entry) ends in a number of low-order Zeros equal to the number of bits in the bit pattern used to index the table.

If the table is indexed by bit patterns which are stored in Process Storage as constants through housekeeping instructions, and, therefore, will never be changed (e. g., these constants will be used as indicators and moved to predetermined locations when certain conditions arise, and later in the program used as modifiers to obtain the desired entry), the number of entries should correspond to the number of bit combinations possible between, and including, all Zeros and the bit pattern which is equal to the highest binary number in all of the used bit configurations. Example: if a four-bit pattern is used, and the combinations are 0010, 0011, 0101, 0111, 1000, 1111 the table must

consist of 4 words (16 characters) and must be located in four consecutive word locations in Process Storage, and the 16 bit address of the "A" character of the first word must end in four Zeros. The corresponding entries must be stored in C and D character fields of the first, in B and D fields of the second, in A field of the third and in D field of the fourth word. The remaining 10 character fields will not be used in this specific table lookup operation. Figure 13 shows the same arrangement in a schematic format.

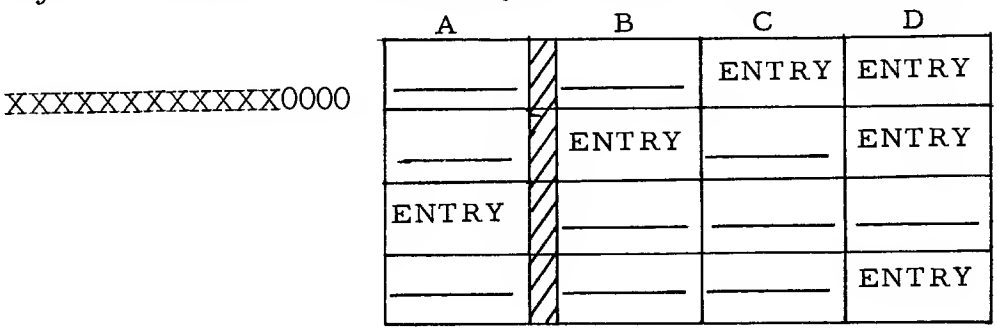


FIGURE 13: The Organization of Character Tables.

The table may be constructed on the same principles when one or more binary numbers are assigned to each of the areas for the Area Scanning schedule. The "scanning round" begins with the highest number. The last used number is always decremented to provide a basic modifier which leads to the next area to be scanned. As the first step, the "shift left two" table will be accessed. The number of table entries should be equal to the total of numbers between all Zeros and the highest number assigned to the areas. Example: if 80 numbers are assigned from 79_{10} to 0_{10} (117_8 to 0_8) a 20 word table is needed for "shift left two" table and the A character of the first word must have a PS address ending in seven Zeros. In this case, obviously all character fields of the table will contain entries. The area identifier must be reset to the highest number used after each "scanning round" has been completed (i. e., the area identifier is decremented to all Zeros). A good method of doing this is to avoid assigning an area to the all Zero pattern and keep the address of the reset routine in the first table entry of the Program Branch Table. When a BRA* instruction is executed, modified by the shift left version of the all Zero identifier, the reset routine will be executed. By this method, no test for Zeros is necessary after each decrement. If a character table is indexed by a code bit pattern, which is obtained during input operation (from channels) or from the Processor, and it is possible to acquire other than predetermined valid patterns, the construction of the table must be different. Since all possible combinations of the indexing bit pattern may occur as indexing factors, the table (i. e.,the number of characters in the table) must consist of 2^x characters,where x = the number of bits in the indexing bit pattern. For valid indexing patterns, the desired equivalents will be stored as entries; for the unused (invalid) combinations (or sometimes for

combinations the equivalents of which are desired to be deleted) all Zeros or all Ones will normally be stored to facilitate transfer to a different routine should such an entry be obtained from the table.

If the entries are full words, or are the contents of Address (and Limit) fields in words, they must be accessed by word addresses. In such cases, modifiers accessing the proper table entries may be obtained from a predetermined reference location or may be derived from a "shift left two" table indexed by a basic modifier.

In a table accessable by word addresses, the 16 bit address of the first entry in the table must end in a number of Zeros equal to the number of bits in the modifier. Consequently, if the word table modifier is derived from a "shift left two" table which is a character table, each entry must be longer by two bits than the number of bits in the bit pattern with which the shift table is indexed. A "shift left two" table should be organized in a way that provides the most convenient application for the programmer. When executing other than CH instructions, the machine logic simply ignores the two low-order bits of the operational PS address and accesses the word which is represented by the 14 high-order bits of a 16 bit address. Therefore, an entry in a "shift left two" table does not necessarily have to contain two Zeros in the two low-order bit positions. It may contain any combination of bits (i. e., 00, 01, 10, 11), and the instruction, of which the operational address is modified by the entry, will be executed on the same word. Example: a word in a Program Branch Table contains an Address, a Limit (i. e., another address), and a reference item in the D field. The entry in the shift table should contain two Ones in the two low-order bit positions. The same entry can then be used as a modifier to access any of the three fields in the table word. If it is used with a CH instruction, the D field is accessed; with another type of instruction the Address or Limit field or the entire word can be accessed. FIGURE 14 shows the organization of the "shift Left Two" Table, indexed by a pattern of 5 bits.

XXXXXXXXXXX00000

A	B	C	D
00000000011	00000000111	00000001011	00000001111
00000010011	00000010111	00000011011	00000011111
00000100011	00000100111	00000101011	00000101111
00000110011	00000110111	00000111011	00000111111

FIGURE 14: Organization of a "Shift Left Two" Table

The examination of the table clearly shows that, if a table is indexed by a character (between 1 - 11 bits), the entries of the consecutive character fields contain consecutive word addresses because the 14 high-order bits of each consecutive entry is one higher than the preceding entry (i. e., 4 character locations higher).

A table entry is accessed by the program through address modification. The indexing pattern must be loaded into the low-order bit positions of a register defined in the R and S field of an Instruction Word. The operational address of another instruction (the contents of the Address field in the Instruction Word, "W") which operates on the table entry must be modified by this indexing pattern. Therefore, the same register and bit number, given as R and S for loading the indexing pattern, must be defined as M and L in the latter Instruction Word.

The unique organization of the Normal Mode Program is introduced in Section V-6. A Program Branch Table controls the operation on an enter-exit basis. Each time the program enters an area, by first decrementing (or resetting) the last used area identifier and then using the shifted left two version of the area identifier as modifier, the associated word in the table will be accessed. The Address (bits 47-32) in the table word provides the location of the first instruction of the program routine to be executed. When leaving an area, the initial address of the routine to be executed when the program returns to this area must be recorded in this table word by the program, unless it is not changed.

One method of organizing the table is to record the initial address of the selected routine in the Address field of each table word and access the routine by a BRA* instruction. The operational address of this instruction must be defined in the source program as the start-address (i. e., the address of the first character of the table, normally referred to by a label) modified by ML containing the shifted left two version of the area identifier. If an area is represented in the table by more than one entry (word), through multiple area identifier assignment, the next of these entries referring to the same area in the programmed scanning sequence must be updated when the program leaves the area.

Another method for organizing a Program Branch Table in which each of several areas is represented by a group of entries, would be to always update (if updating becomes necessary) the same entry of a group and upon entering the area to access the updated entry through the other entries of the group; except when the updated entry itself is accessed. The entry which is always updated must be a BRA instruction. The rest of the entries for a group must be assembled as instructions utilizing the recognition of a modified asterisk as an address, by the Assembly Program. As a result, the Address field of the Instruction Words (bits 47-32) will contain the location address of the updated entry. These assembled instructions will never be executed since the contents of the Address field (W*) are used to access the updated entry trhough a BRA* instruction. Therefore, once the program is loaded

into Process Storage, the programmer may utilize the Limit and D, or C and D fields of these entry words (except the updated entry) for storing reference items. This method results in a double branch operation when one of these words is accessed (i. e., 1. BRA* to updated table word. 2. Branch to due program routine), and in a single branch (BRA*) operation, when the updated table word is directly accessed. However, the program must recognize the condition that only one table entry per area must be updated and which one.

Example: Seven Words represent the same area in a table. It is decided to update the first in sequence each time the program is leaving the area. The remaining six words must be coded as instructions, each containing an Address (bits 47-32) of *-N, where N = 4 times the number of word positions that the constantly updated table word is located lower in sequence in the table. FIGURE 15 shows the schematic presentation of this example.

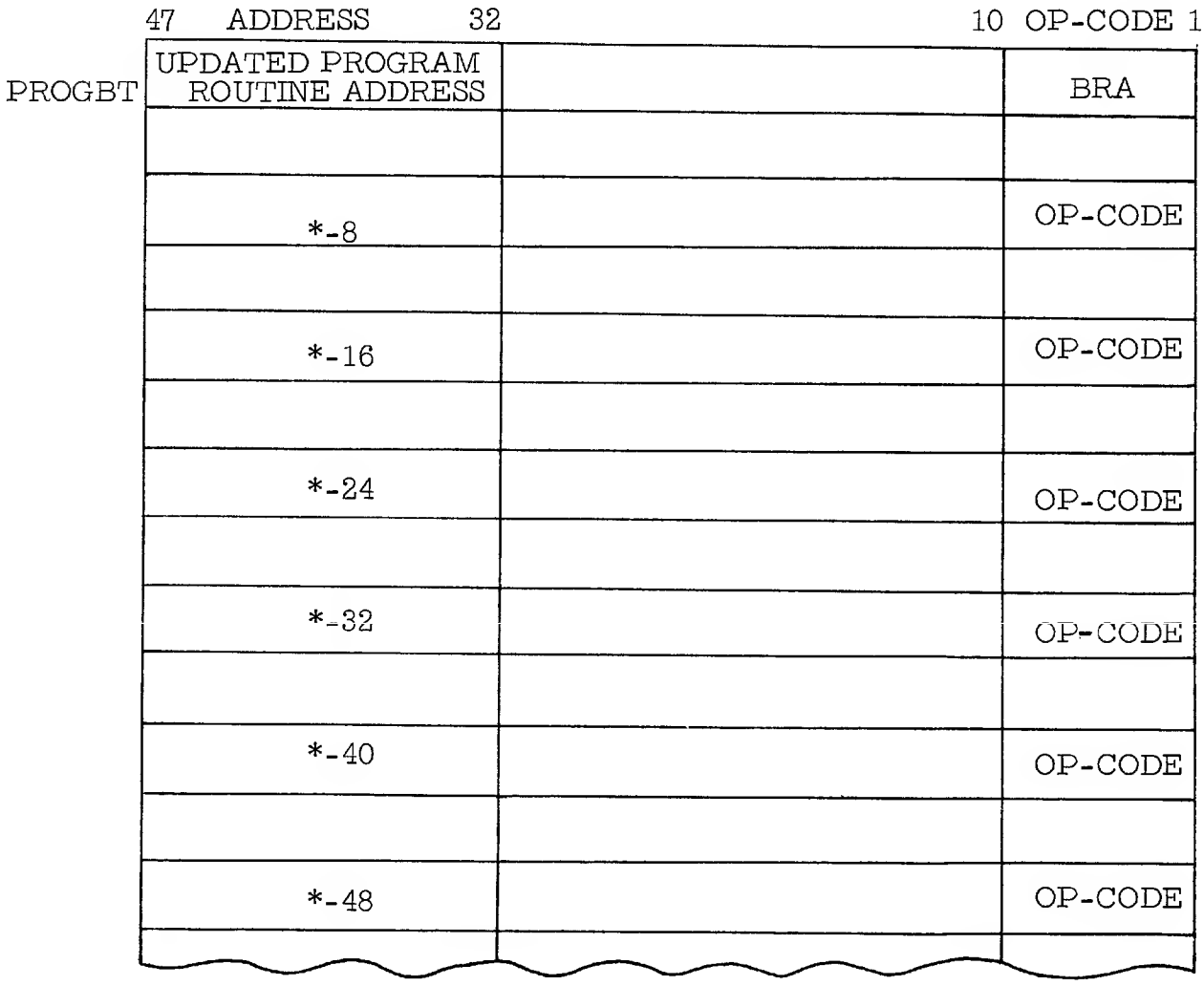


FIGURE 15: Organization of Program Branch Table for Simplified Updating.

The disadvantage of this second organization method is that cross-reference items may not be kept in the updated table words (i. e., in the Limit and D, or C and D fields) since every one is an Instruction Word.

A table may serve more than one purpose. For example, in some applications, the same table may be used for validity check and for code conversion. Only one access per table entry is needed to perform the double function. In some cases, multiple table lookup operation must be used in order to operate the program in the most efficient way. FIGURE 16 is a coding example and FIGURE 17 illustrates the Table organization in Process Storage for an example of multiple table lookup operation combining both above-mentioned methods.
(For FIGURE 17 see next page.)

NAME	OP CODE		R	S		F		ADDRESS, ML	COMMENTS
	LOI*		Z	6				IPLWD	LOAD INPUT CHARACTER
	BRA							UPDTE	EOB ROUTINE
	LOD		Z	7				ATABLE, Z6	LOAD CONVERTED CHAR., OR ZEROS IF INVALID
	BRZ		Z	7				INVALD	INVALID CHAR. FOUND
	LOD		Y	6				BTABLE, Z7	LOAD "ACTION MODIFIER"
	BRA*							CTABLE, Y6	BRANCH TO PROPER ROUTINE

FIGURE 16: Coding Example for Multiple Table Lookup Operation.

Explanation: Table "A" is indexed by a 6 bit input character. The Input Area is controlled by a Limit Word, the location of which is represented by a Label (Name) IPLWD. Each entry in the table contains a 7 bit character (i. e., another code representation of the same character) for valid, and Zeros for invalid input bit configurations. Table "B" is a "shift left two" table indexed by the 7 bit valid characters obtained from Table "A". Each entry is a 6 bit modifier used to access the proper entry in Table "C". It is organized on the principle introduced before to provide access to all three fields of the table word in Table "C". Table "C" is an Action Branch Table, which should not be confused with the Program Branch Table used in "area scanning". Each entry determines the routine to be executed depending on the currently processed valid characters. If a data character is acquired from Table "A" (in the example, 0001000), the obtained modifier from Table "B" (000011) leads to the regular character processing routine (e. g., storing the converted character in a process area). If a special character is acquired (in the example, 1010110), the modifier (011111) leads to the special routine to be executed upon finding this specific special character. Reference items pertaining to each routine may be kept in the Limit and D fields of the words in the Action Branch Table if necessary.

A: Table for Code-Conversion and Character Validity Check

A TABLE =

	A	B	C	D
XXXXXXXXXX000000				
XXXXXXXXXX000111				00000000000
XXXXXXXXXX001010			00001010110	
XXXXXXXXXX001100	00000001000			

B: Table for Action Modifiers

B TABLE =

	A	B	C	D
XXXXXXXXX0000000		00000000011	00000000011	00000000011
XXXXXXXXX0001000	00000000011			
XXXXXXXXX1010110			00000011111	

C: Action Branch Table

C TABLE =

	ADDRESS	LIMIT	D
XXXXXXXXXX000000	ADDRESS OF CHAR. PROCESSING ROUTINE	REFERENCE ITEM 1	REFERENCE ITEM 2
XXXXXXXXXX011111	ADDRESS OF SPECIAL ROUTINE	REFERENCE ITEM 1	REFERENCE ITEM 2

FIGURE 17: Utilization of Multiple Table Lookup Including a Double Purpose Table

These items may be accessed through Get Limit (LIMIT), or CH(D) instructions by using the same modifier obtained from Table "B".

The number of possible table applications in a program is unpredictable. Whenever the programmer can efficiently implement a table lookup operation in his program, he may do so.

5. DECREMENT, INCREMENT, COUNT-CHECKING

A. DECREMENT. Register Z is an 11 bit count-down counter. A binary number located in one of the character fields of a Process Storage word can be decremented by one by using two instructions. The first instruction must load the number into Register Z and the second one, flagged to "decrement Z Register", must unload it to the original, or another specified, character location. Since the decrement takes place before the execution of the instruction so flagged, the second instruction will unload the already decremented number. An example is given in FIGURE 18.

OP CODE	R	S	F	ADDRESS, ML
LOD	Z	1 1		CHAR
UNL	Z	1 1	1	CHAR

FIGURE 18: Decrement of a Binary Number

Explanation: For the decrement itself, once the number is located in Register Z, no independent instruction is necessary because any instruction can be flagged to decrement the contents of Register Z as an additional function. When decrementing the contents of Register Z many times in a loop, it is not necessary to unload Z each time it is decremented, provided that Z is not used elsewhere in the loop.

If Register Z contains all Zeros when the decrement is performed, the result will be all Ones (i. e., the machine logic treats it as the twelve bit number, 4000_8, before the decrement; it becomes 3777_8 after).

NOTE: BRZ, BRZ*, BRO, BRO* instructions, if so flagged, are similarly executed after the decrement; however, the condition on which to branch (i. e. S number of low-order bits in R register) is tested before the decrement. Example: the content of Z register is 1_8, the instruction to be executed is BRZ Z 11 1 W; branch will not take place because Z register did not contain zeros at the time the test was made.

B. INCREMENT. There are two methods available to increment a binary number by one. One method is to use Register Z, in which case the highest possible number to be incremented without the loss of carry is 3776_8 (2046_{10}). Two instructions are necessary. The first instruction must load the Ones complement of the number into Register Z, and the second one, so flagged to "decrement Z Register", must unload the recomplemented decremented complement, which is equivalent to the original number plus one. An example is shown in FIGURE 19.

OP CODE		R	S		F		ADDRESS, ML
LDC		Z	1	1			CHAR
ULC		Z	1	1	1		CHAR

FIGURE 19: Increment of a Number by Using Z Register

The second method is to use the Address field (47-32) of a Process Storage word as an increment counter. The highest possible number to be incremented without the loss of carry is 177776_8 (65534_{10}). The basic number to be incremented by one (successively) must be stored in the Address field of the selected Process Storage word, through the Operational Address Register. Only one instruction is necessary each time the number is to be incremented, a "Load and Increment" (LOI) instruction without naming a Register R or size S, or either one, or defining one or both as Zero. An example is shown in FIGURE 20.

NAME	OP CODE		R	S		F		ADDRESS, ML	COMMENTS
	LOD		X						SET X TO ZERO
	LOD		Y						SET Y TO ZERO
	PTA							COUNT	MOVE ALL ZEROS TO THE INCREMENT COUNTER
	LOI							COUNT	RESULT: 0000000000000001

FIGURE 20: Straightforward Increment of a Binary Number

Explanation: COUNT is label (Name) of a PS word, defined as a counter.

If it is necessary to check whether the incremented number has been increased to a predetermined value or not, this value must be stored in the Limit field of the same Process Storage word, again through the OAR

and a "Compare Address to Limit" instruction must be executed following each increment. If the two numbers become equal, the instruction immediately following the CAL instruction will be executed; if not, it is skipped. The increment may be done after the comparison by the application of a single "Compare Address to Limit And Increment" (CAI) instruction, in which case the LOI instruction is unnecessary. However, in this case, when the next instruction in sequence is executed as the result of an equal compare, the Address field will contain a number, one higher than the Limit field, in the tested word.

The second introduced method of increment can be well utilized when changing an Address in a Limit Word after EOB condition has been detected. When the 11 high-order bits of the BCC are no longer needed, an LOI instruction can be executed to change the five low-order Ones to Zeros. The new block address then can be moved in the "A" field by treating the Limit Word as a Data Word.

C. COUNT-CHECKING. There are cases when it is necessary to check whether an updated count is less than or not less than (i.e., equal or higher) a predetermined number. One important use is for checking the block-count of the available memory space on a programmed schedule in order to maintain a safe input operation. If this block-count falls below a predetermined number, the output must be increased and the input decreased or even temporarily stopped by the program. This may be done by changing the polling schedule, or suspending Polling altogether, until the available chain again reaches the required minimum number of blocks. It may even affect the traffic between the IBM 7750 and the Processor if the program specifications contain such provisions. This type of checking may be done by testing only the high-order bit positions of the updated count.

   Example: 64 blocks are required by an application as a safe available buffer space minimum. Since $64_{10} = 100_8$, at least one of the five high-order bits of the updated 11 bit block-count must be a One bit in order to have at least 64_{10} blocks available at the time of checking. A mask composed of the following bits: 11111000000, must be kept as a constant in a Process Storage character location to perform the check. If the mask is AND-ed together with the updated block-count and the result is all Zeros, this means that the block-count is less than 64_{10}. The coding of this example is shown in FIGURE 21.

OP CODE		R	S		F		ADDRESS, ML	COMMENTS
LOD		Z	1,1				AMSLWD+3	BLOCK COUNT INTO REGISTER Z
AND		Z	1,1				MASK	AND MASK TO REGISTER Z
BRZ							CHANGE	BLOCK COUNT LESS THAN 64
								CONTINUE ORIGINAL PROGRAM ROUTINE

FIGURE 21: Count-Checking.

SECTION XII

ERROR PROCEDURE FOR CHANNEL AND PROCESS CONTROL ERRORS

1. CHANNEL ERRORS

Errors occurring between the character assembly/distribution area of the Process Control (i.e., the Assembly Area of a Channel Word when it is in the Control Storage Data Register) and a remote device operating on one of the IBM 7750 Process Control Channels are called Channel Errors (see FIGURES 1 & 22). The possible Channel Errors are listed below. The detection of Channel Errors is automatic. However, it is not enough to detect these errors; quick corrective and recovery procedures initiated by the IBM 7750 program must be performed. Therefore, the type of the error and the Communication Channel on which the error has been detected must be indicated to the program. Furthermore, a special method must be used to secure immediate access for the program to examine the situation, identify the error and the channel, initiate quick action for the error correction according to the found condition, and, when necessary, reroute the traffic of the erroneous channel (i.e., direct the transmission through other channel(s) to other Terminal(s)), as predetermined in the system specifications, or switch the operation to the stand-by IBM 7750 in order to maintain the continuous operation of the system.

The presently applicable components that may be attached to the 7750 (i.e., the available types of Terminals, Subsets, Channels) may produce four different types of high-speed Channel Errors and two different types of low-speed Channel Errors. In order to understand the following descriptions, note that according to the present configurations, a full-duplex high-speed Communication Channel will operate through one High-Speed Channel Adapter and two Process Control Channels (scanning points).

The listing and brief description of the Channel Errors and the programmed preparations necessary for the detection of such errors are as follows:

A. HIGH-SPEED CHANNEL ERRORS AND PREPARATIONS FOR THEIR DETECTION

1. Interlock Error - This is caused by the lack of power on the Subset; it may be turned off or may be lost by technical failure. This error may occur on any type of Subset regardless of the status, Send or Receive, of the channel. The channel may be either half-duplex or full-duplex. No programmed preparation is required for the detection of this type of error.

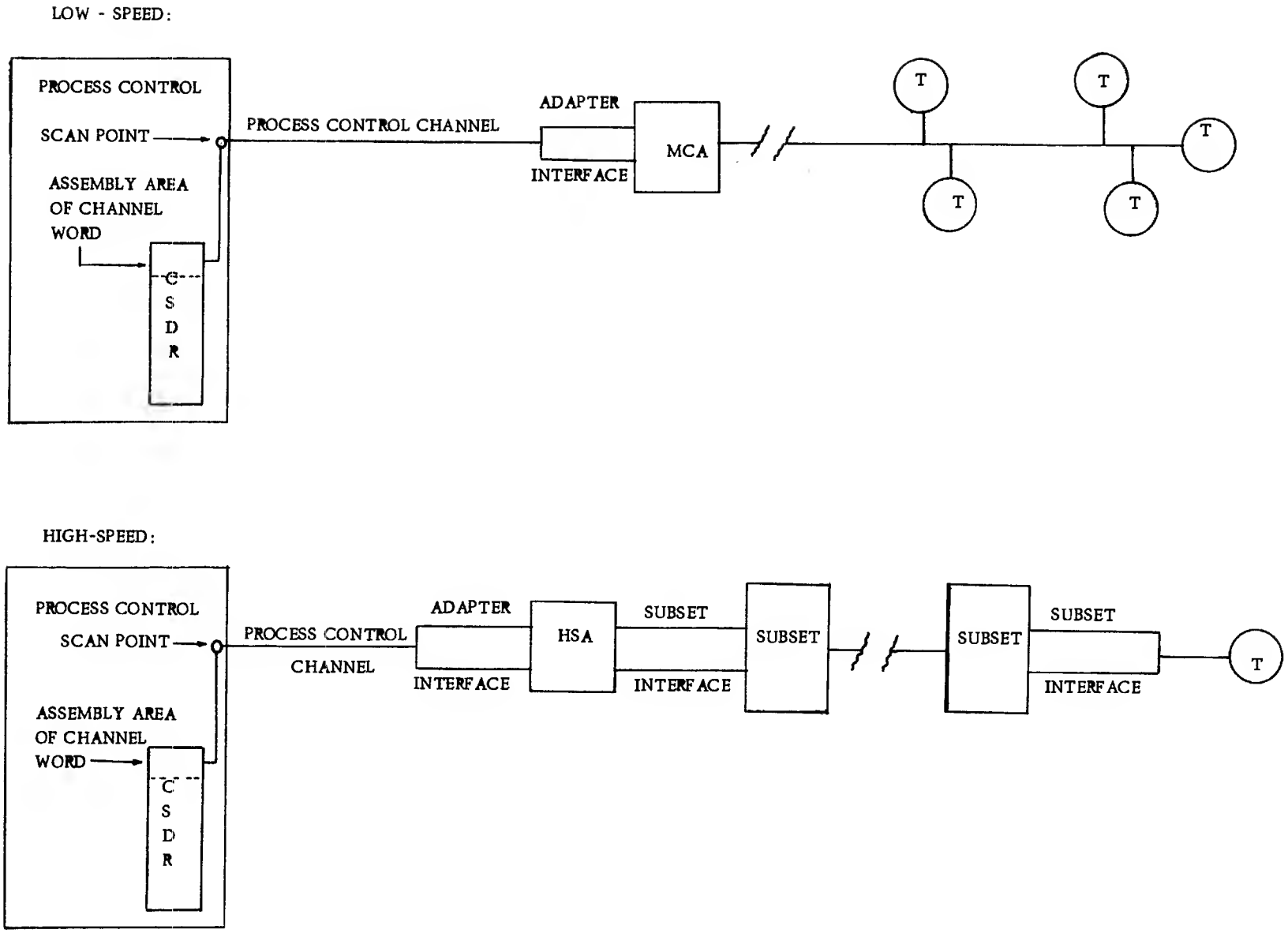


FIGURE 22: Systems Locations of Possible Process Control and Channel Errors

2. Carrier On Error. - This type of error may occur during operations with high-speed channels functioning with certain (other than FM) type Subsets in Receive status only (i.e., on half-duplex channels when in Receive status, and on the Receive line of a full-duplex channel.)

Carrier is the current, whose frequency (which is above the voice frequency) is modulated by signal frequencies representing the binary information to be transmitted. The modulated carrier is transmitted over the circuit (e.g., telephone line) to the receiver, where a demodulating circuit recovers the original binary information.

On the Digital Subset, there is a line named Carrier On which rises a nominal period of time after the Subset receives Carrier and falls over a nominal period of time after the Subset no longer receives Carrier.

The Interface of such a Digital Subset is connected to the Adapter Control Interface of the IBM 7750 by a High- Speed Channel Adapter of a different type than HSA2.

The automatic Carrier On Error detection is an optional feature. If the system specifications do not require automatic Carrier On Error detection on an application, the program must set the Control Bit (bit 13) in every Receive and Send Channel Word assigned to the channels functioning with such Digital Subsets, to logical Zero in the housekeeping routine, and it must not be changed throughout the operation.

If the feature is required, the setting and resetting of the Control Bit in the effected Synchronous Receive CWD would provide a means for the program to command the logic for the desired functions. In this case, the Control Bit should be turned on just prior to the reception and turned off when the last character is received. By setting the Control Bit to One, the Carrier sensing circuitry is activated in the High-Speed Channel Adapter; if the error occurs, it can and will be detected and indicated. However, if the Control Bit is not turned off at the end of a reception, the Carrier sensing circuitry remains active and, although there is no existing error, an error will be indicated each time the channel is scanned because the Carrier fell at the end of the transmission, due to normal conditions.

In order to avoid the constant setting and resetting of the Control Bit, another method has been developed whereby the Control Bit is set to One and not changed. The description for this is as follows.

The Channel Adapter detects Carrier On in Receive Status only. The conditions for indicating an error are: Receive, Carrier Off, Control Bit set to a One, and a Character Complete (Adapter Control Interface) from Process Control. From the above conditions, it can be seen that if a CWD is in Hunt status with the Control Bit set to a One, a Carrier On error cannot be detected until after the CWD has established character synchronization and at least one character has been received. Since the Normal Mode Program would not be able to control the conditions necessary for checking or to stop checking the error on the required schedule, the Action Delay Feature can be used to perform this function. The procedure is as follows: The CWD is first placed in Hunt status with the Control Bit set to a One. This in effect commands the Channel Adapter to start checking for the Carrier On error after receiving the first valid character. The program prepares the Y field associated with the channel in the following manner: Bit 1 must be set to Zero, (Send-Receive Bit) to maintain the receive status of the channel. Bit 2 (Control Bit) is set to a One. Bit 3 (Synchronous Bit) is set to a One. 16_8 must be placed in bit positions 8-4 (pattern for Hunt status), a One must be placed in bit position 9 (Action Delay Bit) to assure the continuous request for the recognition of the Action Delay character; 3 minus the desired number of character delay times must be placed in bit positions 10 and 11 (Sequence Counter). After Character Synchronization has been established and the first character received, the Carrier On checking begins. One of the last four characters to be received on the channel must be the Action Delay Character (i. e., depending on the specifications of the specific application; however, the Sequence Counter must be set to correspond with the positioning of the ADC). Recognition of the ADC will automatically place the CWD back in Hunt status after the determined number of character times (i. e., 0 to 3), by loading the Y field shown above into the channel word, thus stopping additional Carrier On checking.

The described procedure can be performed for the CWD on a half-duplex channel or for the CWD of the Receive line on a full-duplex channel each time a reception takes place. The system specifications must require that each incoming transmission end with an Action Delay Character followed by none or no more than three information characters.

The Control Bit in the CWD of the Send line of a full-duplex channel has no meaning. In the CWD of a half-duplex channel, the Control Bit normally will have the correct setting, logical One, when a Send operation begins, because the only time the checking begins is after the first character is received in a Receive operation.

Whether or not the system specifications will require the use of the automatic error detection feature will probably depend upon how rapidly it is necessary to respond to such an error condition.

3. Time Out Tag Error (3 seconds) - This error condition may occur during operations with high-speed channels, functioning with FM Subsets only, if the first character is not received within a predetermined time period after the IBM 7750 initiated an incoming transmission.

The conditions necessary to detect this type of error are established differently for a half-duplex than for a full-duplex channel. In addition, on a full-duplex channel, programmed intervention is necessary after the first transmitted character has been received to avoid a possible error indication when conditions are normal. Therefore, the IBM 7750 program has different responsibilities for handling the operation of a half-duplex and of a full-duplex channel.

The Interface of an FM subset is connected to the IBM 7750 Adapter Control Interface by a Type 2 High-Speed Channel Adapter (HSA2).

On a half-duplex channel, the error condition arises if a "Character Complete" condition (i.e., the reception of a complete character from the channel) is not recognized by the Channel Adapter within three seconds after a hardware device, called the Hold Latch, in the Channel Adapter is turned off. This three-second time interval is controlled by a hardware counter. When a "Character Complete" condition is recognized, the Hold Latch is automatically turned on and the time counter is reset by the hardware. The Hold Latch is turned off automatically upon the change of the status of the channel from Send to Receive. Therefore, no programmed function is required to establish, or to eliminate, the conditions necessary for the error detection. Obviously, the change of the status from Send to Receive is always a programmed function (normally by the application of an SCC).

On a full-duplex channel, the error condition occurs if three seconds elapse from the time the Hold Latch is turned off. The Character Complete line in the Adapter Control Interface is not tested because it has a reduced function when a full-duplex channel is serviced by the Channel Adapter. The program must set the Control Bit in the Send CWD to logical Zero. This will cause the automatic turn-off of the Hold Latch and, as a consequence, the start-off of the time counter. The Control Bit may be set to Zero through the utilization of the Action Delay feature. The program prepares the Y field to contain a Zero in bit position 2, which corresponds with bit position 13 in the CWD. The Action Delay Bit must be turned on, and the ADC must be stored in the output area. The Sequence Counter can contain any desired number from 0_8 to 3_8. Upon the recognition of the ADC in the Assembly Area,

the Action Delay Bit will be turned off, the Sequence Counter incremented. The Y field is loaded into the CWD when the Sequence Counter is incremented to 08. The Control Bit becomes a logical Zero, and the Hold Latch will be turned off. This starts the Counter. Bits 11-10-9 may permanently be stored in the Y field, to prepare the Action Delay operation for the next consecutive time. The error condition will be indicated when the time is out in the Counter. Therefore, the Normal Mode Program must check the input area of the channel, and when the first received character is found, set the Control Bit in the CWD of the Sending line to logical One. This may be done by the execution of an MOC instruction. The Control Bit being a logical One, the Hold Latch is automatically turned on and the time counter is reset by the hardware. However, this must be done within three seconds from the time the count started; otherwise, the hardware counter in the Channel Adapter will count down and an error condition will be indicated when an error has not occurred.

4. Data Transfer Error - This type of error may occur while operating with half-duplex and full-duplex channels functioning with any type Subset, but in Send Status only.

   Each time a bit is sent to the Channel Adapter from the low-order bit position of the Assembly Area, it is shifted into the Transfer Check Bit position (bit 23) of the CWD. When the bit is sent from the Adapter to the channel, the Adapter sets a line in the Adapter Control Interface (i. e. ,Data In line) to the value of the transmitted bit. When the next bit is requested by the channel, the two values, the value of the Data In Line and the Data Transfer Check Bit, will automatically be checked by Process Control, and if they do not compare, an error is indicated.

   No programmed preparation is necessary to detect this type of error; the procedure is fully automatic.

B. LOW-SPEED CHANNEL ERRORS

1. Data Transfer Error - This is the same type of error as the Data Transfer Error for high-speed channel operations; its detection occurs in the same manner as described in Chapter A-4. of this Section. It can only be detected on channels in Send Status.

2. MCA Parity Error - This type of error can be detected while operating with low-speed channels, both in Receive and Send

Statuses. The Multiplexing Channel Adapter has a memory. If there is more than one Multiplexing Channel Adapter in a system (up to four), they share the same memory. In this memory, an 11 bit control character is assigned to each channel connected to any one of the Channel Adapters. Nine bit positions in a control character are used to control the bit operation of a specific channel; one of them is the one bit data buffer for the channel and one bit is the Parity Bit. This bit's function is to provide odd parity. The memory has a Buffer Data Register, the size of which depends on the number of MCA's in a system; consequently, it may be an 11, 22, 33 or 44 bit position register. Each time an MCA scans one of its lines, the associated line control character is read out from the memory into this register. If there are more than one MCA in a system, their scanning activity is synchronous (i. e., each scans the corresponding scanning points at the same time). If the number of lines is not identical on all MCA's, the scanning schedule is adjusted to the one having the largest number of lines. In such a case, some MCA's may scan "dead scanning points" (i. e., scanning points which have no connection to working lines). Therefore, if there are four MCA's, in each MCA scanning cycle (which is 14u), the four associated control characters will be read out of memory into the 44 bit position Buffer Data Register. If a parity error (i. e., even parity) is detected in a control character in the Buffer Data Register on a matched scan cycle (i. e., when the CWD assigned to the same channel whose control character contains the parity error, is located in the IBM 7750 Control Storage Data Register), the error is indicated through the proper Channel Error procedure.

No programmed preparation is necessary for the detection of this type of error.

C. THE DETECTION AND INDICATION OF CHANNEL ERRORS

With the exception of the Data Transfer Error, which is detected by the Process Control of the IBM 7750, all other Channel Errors are detected by the Channel Adapters. The detection and following indication of all Channel Errors is fully automatic; in some cases, however, programmed preparations are required to establish the conditions necessary for automatic error detection as described before. With the exception of the Time Out Tag Error, upon the detection of a high or low-speed Channel Error, the Channel Check Light corresponding to the Process Control Channel, through which the erroneous channel is scanned, will be turned on the Operator's Panel.

The Indication of Channel Errors happens as follows:

1. Error Indication for High-Speed Channel Operations

Each high-speed channel, half-duplex or full-duplex, has two Error Channel Words (ECWD) assigned in Control Storage with predetermined locations. The program must set the Error Channel Words according to the specifications given in Section III-1, C. Each ECWD is preassigned to a specific error condition. Since the Control Storage address of an ECWD identifies the channel in error and the type of error, as soon as the program has access to it, the necessary corrective procedure can be initiated.

There are four possible high-speed Channel Errors as listed and only two Error Channel Words per high-speed channel. However, the description of errors shows that only three types of errors may be possible for a high-speed channel, depending on the type of Subset with which it is associated, because the Carrier On Error may occur only on channels functioning with certain Subsets and the Time Out Tag Error may occur only on channels functioning with FM Subsets. However, one of the three remaining errors, the Data Transfer Error, which may occur on channels being in Send Status only, is indicated by the hardware in the regular Channel Word by setting the Not Error Bit (bit 24 in the Synchronous Send Channel Words) to logical Zero, which is the error status for this bit position. For each of the remaining two errors, a specific Error Channel Word represents a certain type of error.

When an error, the type of which has an ECWD assigned, is detected, the High-Speed Channel Adapter accesses the proper ECWD instead of the regular CWD for that channel, and also sets the Not Error Bit (bit 13) in the ECWD to logical Zero. When the machine logic recognizes the error condition (i. e., logical Zeros in bit positions 13 and 14 of the CSDR = a logical Zero in the Not Error Bit position of a Start-Stop Channel Word), the Channel Error Bit (bit 8) is turned on (i. e., set to logical One) in the Channel Service Register, the Control Storage address of the ECWD is placed in the seven low-order bit positions of the Channel Service Register, and Service Mode is requested in the Mode Request Register. Control will be turned over to the Service Mode. If the Service Mode or Channel Service Mode has previously been requested, the Service Mode may be requested, and the address of the ECWD may be placed into the Channel Service Register only when the ECWD of the erroneous channel is read out into the Control Storage Data Register the next time after the previous request(s) have been terminated.

When a Data Transfer Error is detected, the machine logic turns off the Not Error bit (bit 24) in the Synchronous Send Channel Word, and the procedure is the same as described above; the only difference is that the address of the regular CWD, and not of an ECWD, will be placed in the Channel Service Register. The restrictions for obtaining the service are also the same.

2. Error Indication for Low-Speed Channel Operations

Low-speed channels have no Error Channel Word assigned; Channel Errors are indicated through their regular Channel Words. There are only two types of low-speed Channel Errors. For channels in Receive status, only an MCA Parity Error may occur; however, for channels in Send status, both an MCA Parity Error and a Data Transfer Error may occur, and both are indicated by the same method. Therefore, for low-speed channels in Send status, the program cannot identify which of the two possible error conditions caused the error indication.

In each Channel Word assigned to a half-duplex or to one of the two lines of a full-duplex low-speed channel, bit position 13 is the Not Error Bit. If this bit position contains a logical One (the Not Error Bit is on), the status indicates a "not error" condition; if it contains a logical Zero (the Not Error Bit is off), the status indicates an "error" condition. The stored program sets the Not Error Bit to logical One in each of these Channel Words, and the hardware turns it off upon the detection of an error condition.

When an error is detected, the Not Error Bit is automatically turned off and the machine logic, sensing logical Zeros in bit positions 13 and 14 of the Control Storage Data Register, performs the following functions: turns the Channel Error Bit (bit 8) "on" in the Channel Service Register, places the Control Storage address of the Channel Word into the seven low-order bit positions of the Channel Service Register, and turns the Service Mode Bit on in the Mode Request Register. Control will be turned over to the Service Mode as soon as it is available; the restrictions for not obtaining the Service Mode immediately are the same as those described for high-speed Channel Errors in Chapter C-1 of this Section.

D. THE IDENTIFICATION OF CHANNEL ERRORS

Any type of Channel Error for high or low-speed channels will automatically turn the operation of the IBM 7750 to the Service Mode. The program written for the Service Mode must have efficient routines for

the quick identification of the specific error indicated and for the initiation of the proper corrective actions.

The program must examine the settings of the error bits (bits 11-8) in the Channel Service Register in order to execute the proper routine. Since each of the three high-order bits in the CSR represents one Process Control Error only, and bit 8 may indicate one of several types of Channel Errors, it is quite obvious that the program should first check bit 8. A Branch On Test (BRT) instruction is adequate for the test. The next step would be to examine the contents of the seven low-order bit positions in the Channel Service Register to identify the channel in error and the type of the error. Once the error and the source are identified, the desired action can be initiated without delay.

E. CORRECTIVE ACTIONS FOR CHANNEL ERRORS

There are no set rules for handling Channel Errors; it depends entirely on the requirements of each individual system application and must be described in the system specifications. However, certain possibilities may be pointed out on a general level.

It is certainly not desirable to set an alarm when there is no reason for it. The environment of a network sometimes may be the source of conditions which result in effects recognized as errors when no error has occurred. Therefore, the programmer may follow a principle of "trying out" the error indication first to be sure of the presence of the signaled error; before initiating a complicated error procedure. The way to do this is to eliminate the error indication by resetting the bits (triggers) to "not error" status and to keep a count for each error type, by channel, at certain memory locations. If the indicated channel error has been a "legitimate" error, the Service Mode will repeatedly be requested in consecutive scan cycles for that specific channel after each resetting, and the count will rapidly reach the predetermined number required to execute a certain error routine. At this time, the counter must obviously be reset. Since it is possible that the same "illegitimate" error indication may occur for the same channel in longer time intervals, provision must also be made to reset the counter from time to time in the Normal Program loop to avoid the unnecessary execution of an error routine by the slow accumulation of such "illegitimate" error counts.

Another principle, already emphasized, is to keep the machine operating in the Service Mode for as short a time as possible and only initiate, but not execute, error routines in this mode; to do so would block other

channels from obtaining service for this or any other mode; furthermore, the Processor could not be efficiently serviced.

When resetting error indications in an Interlock, a Carrier On,or a Time Out Tag Error, the Service Mode Program must load the 14_8 bit pattern into bit positions 18-15 of the affected Error Channel Word. This will reset the error condition in the Channel Adapter. Since the transmission has been interrupted by the error detection and indication, the proper Adapter Synchronization bit pattern must also be loaded into the Character Control Field of the regular Synchronous CWD. Most likely, the retransmission of the interrupted message will be required,by the IBM 7750,from the Terminal. Whether or not to do anything with the setting of the Not Control Bit in the regular CWD depends on the characteristics of the type of transmission in each case.

These functions may be performed in an error routine in the Normal Mode Program before restoring the operation of a channel after the error is"reported"to have been corrected, unless the error indication is completely eliminated in the Service Mode Program on the mentioned "try out" basis.

If a "legitimate" error condition is found, the Service Mode Program must inactivate the channel by setting bit position 26 of the Error Channel Word, or the regular Channel Word (depending which one is used for indicating the error),to logical Zero before turning control over to the error handling Normal Mode or some higher requested mode. If it is a Data Transfer Error on a high-speed channel, bit 14 must also be turned off in the regular CWD. If no provision is made for this action, control will be repeatedly turned to the Service Mode for the very same error, until after the error is corrected.

Under certain conditions defined by the system specifications, the operation will be switched over to the "stand-by" 7750. However, this will most likely happen upon the serial detection of Process Control Errors rather than Channel Errors. If the correction of a Channel Error would require a comparatively long time, the traffic of the affected channel most likely will be rerouted. Normally, the Processor will initiate such an order, but, of course, the system specifications and,consequently, the IBM 7750 program must have made advance provisions for these cases.

The Service Mode Program must turn the Channel Error Bit (bit 8) in the Channel Service Register off before the Service Mode bit is turned off in the Mode Request Register. This should be done by the

instruction immediately preceding the one that turns the Service Mode request bit off. If this happens sooner, the contents of the seven low-order bit positions of the Channel Service Register may be destroyed before the Service Mode Program is able to complete the necessary identification procedure. Each time an End Of Block condition is recognized in a Channel Word, the machine logic first examines the status of the Channel Error Bit in the Channel Service Register. If it is off (i.e., logical Zero) and the Channel Service Mode bit is off in the Mode Request Register at the same time, the Control Storage address of the service requesting Channel Word is placed in the seven low-order bit positions of the Channel Service Register, and the Channel Service Mode bit is turned on in the Mode Request Register. This will happen regardless of whether the Service Mode request bit is on or off. Since the Channel Service Mode request bit must be off when the Service Mode is requested upon the detection of a system error, as explained above, only the "on" status of the Channel Error Bit prevents the machine logic from destroying the address of the error-indicating Channel Word or Error Channel Word in the Channel Service Register.

On the other hand, if the Service Mode Program failed to turn the Error Channel Bit off, the Channel Service Mode would be blocked for all channels as long as the bit is on. For other reasons, however, such as a Process Control Error indication (in which case the placement of an address into the Channel Service Register is not associated with the Service Mode Request), the Service Mode would be available; all other modes (Copy, Out, In, Normal) would also be available.

The Channel Error Bit may be turned off by a Load (LOD) instruction naming the Channel Service Register (C or 4) as an R-register; this instruction resets the four high-order bits (bits 11-8) in the Channel Service Register at the same time.

When a "legitimate" Channel Error is detected, the error routine in the Normal Mode will prepare a message; depending on the program specifications, the error message will be either transmitted to a Terminal (located at the same installation where the IBM 7750 and the Processor are operating), or to the Processor and will be typed out on the console-typewriter. In the latter case, the Attention Bit in the Interface Control Register may be set after the error message for the Processor has been prepared.

2. PROCESS CONTROL ERRORS

Errors occurring in the IBM 7750 Process Control are called Process Control Errors.

A. PROCESS CONTROL ERROR TYPES

There are five Process Control Errors, as follows:

1. PROCESS STORAGE PARITY ERROR

Each time a Process Storage word is read out of the memory into the Process Storage Data Register, the Word Parity is checked by the machine logic. If the Parity is even, an error condition is recognized by the hardware. When a word is written into Process Storage from the Process Storage Data Register, the Word Parity Bit is regenerated.

2. CONTROL STORAGE PARITY ERROR

Each time a Control Storage word is read out into the Control Storage Data Register, it is checked for odd Word Parity. If the parity is even, an error condition is recognized by the machine logic. When a word is written into Control Storage from the Control Storage Data Register, the Word Parity is regenerated.

3. FULL WORD TRANSFER ERROR

When a full word is moved from one Storage to another by an SS type instruction, the Word Parity is checked in the receiving-register **CSDR** or PSDR) and compared with what was transmitted. If a difference is detected, an error condition is recognized.

4. OP-CODE PARITY ERROR

Bit 11 in each Instruction Word is the Op-Code Parity Bit. It must be either logical One or logical Zero, depending on the contents of the Op-Code and Flag fields, to produce an odd parity. It is checked by the machine in the Instruction Register before an instruction is decoded. An Op-Code Parity Error is detected when an even parity is found for the Op-Code and the two Flag Bits.

5. CLOCK ERROR

The control of a computer's operation is determined by specific functions to be performed and certain exact timing impulses, provided by an electronic clock. The timing impulses are produced at a rate of one-millionth of a second, or each pulse is a microsecond in duration. A series,or group of these pulses constitute a cycle and represents the amount of time used to perform certain machine operations, or specific functions,as a unit.

The total number of pulses or micro-seconds needed to complete the complex of such function units before repeating the same functions represent a basic machine cycle.

Control Storage performs its scanning functions in 11u and its mode-operating functions in 17u. These time elements are called Scan-cycles and Process-cyles respectively. The Process Storage functions are divided into a 12u Instruction-cycle, during which an Instruction Word is read out of Process Storage and decoded, and a 16u Execution-cycle, representing the time necessary to execute an instruction. (There are different functions in Copy Mode.) The total length of the two cycles in both Storages is equal to 28u; therefore, the basic machine cycle of the IBM 7750 is 28u. (The CS and PS cycles are overlapping cycles.) The clock controlling the IBM 7750 operation is a 28 stage ring. The lack of a bit,or the presence of two adjacent bits,in the clock represents a Clock Error. If the clock does not function properly, no prescheduled machine operation is possible.

B. THE DETECTION AND INDICATION OF PROCESS CONTROL ERRORS

Process Control Errors are automatically detected by the machine logic. No programmed preparation is necessary for the detection of Process Control Errors.

Three types of Process Control Errors are automatically indicated by the machine logic through the setting of one of the three high-order bit positions (bits 11-9) of the Channel Service Register. Bit 9 for a Process Storage Parity Error, bit 10 for a Control Storage Parity Error, and bit 11 for a Full Word Transfer Error are set to logical One in the Channel Service Register. The Check Switch on the Operator's Panel may be set to either of two positions: Stop or Service Mode. If it is set to Stop, the machine will stop upon the detection of an error; if it is set to Service Mode, this mode will be requested in the Mode Request Register when a logical One is sensed in one of the mentioned three high-order bit positions of the Channel Service Register. Normally, the Check Switch will be set to Service Mode.

An Op-Code Parity Error or a Clock Error will stop the machine regardless of the setting of the Check Switch.

C. THE IDENTIFICATION OF PROCESS CONTROL ERRORS.

The Service Mode Program will identify the indicated Process Control Error through the checking of the three high-order bit positions (bits 11-9)

of the Channel Service Register. These bits must be checked one at a time for a logical One. Branch On Test (BRT) instructions are adequate for this checking.

D. CORRECTIVE ACTIONS FOR PROCESS CONTROL ERRORS

Upon the execution of the Branch On Test instruction, when the tested bit is found in error status, only a short routine should be executed in the Service Mode. The programmer may follow the same principle which has been introduced in Chapter 1-E of this section, and simply eliminate the error indication. A count should be kept for each type of error. When the count reaches a predetermined number, an error routine should be executed in the Normal Mode. However, the Customer Engineers responsible for the maintenance of the machine should be notified of the type and number of the detected errors, regardless of the time intervals at which they occur. Therefore, the counts should not be reset in the Normal Program loop; the error routine should be executed each time a count reaches the predetermined number.

The error bit in the Channel Service Register must be reset by the program before leaving the Service Mode. The execution of a LOD instruction, naming the CSR (C or 4) as Register R, resets all four high-order bits in the CSR. The Service Mode routines should be kept as short as possible. The error routine in the Normal Mode will be similar to the one followed in case of Channel Errors.

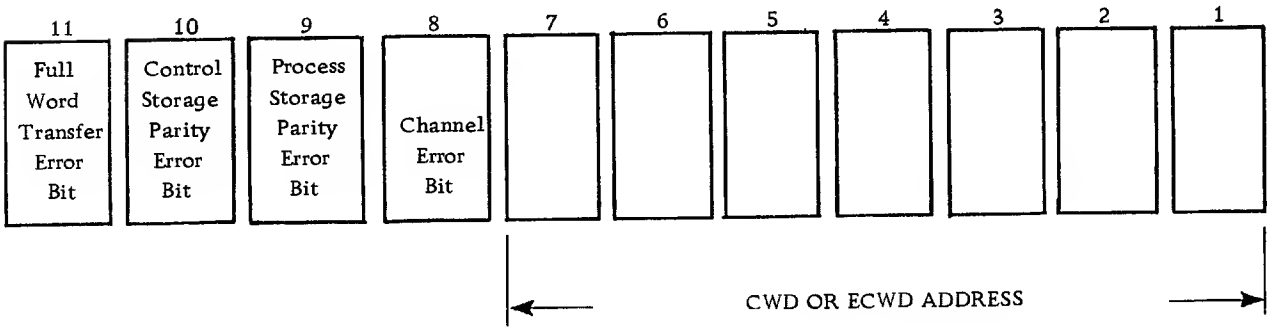
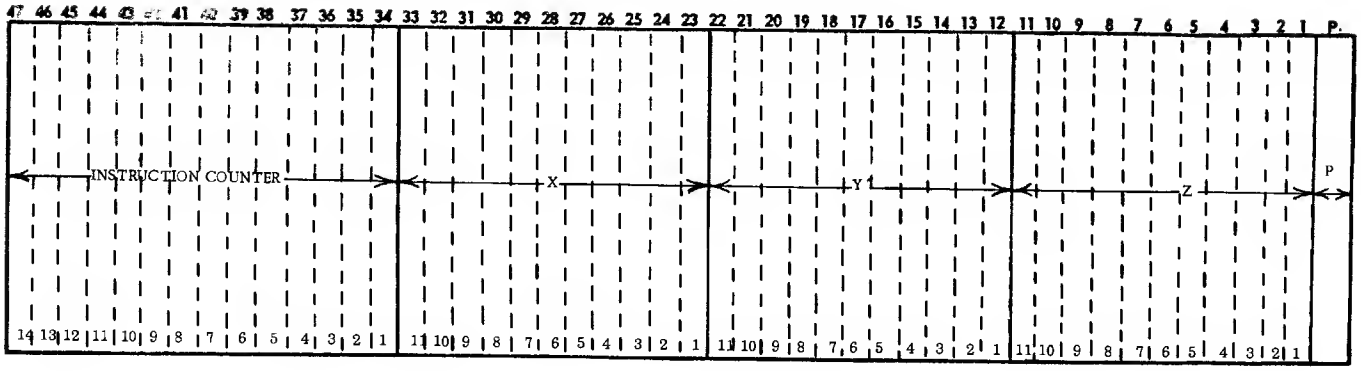


FIGURE 23: The Channel Service Register

APPENDIX I - CHART 1

PROCESS WORD

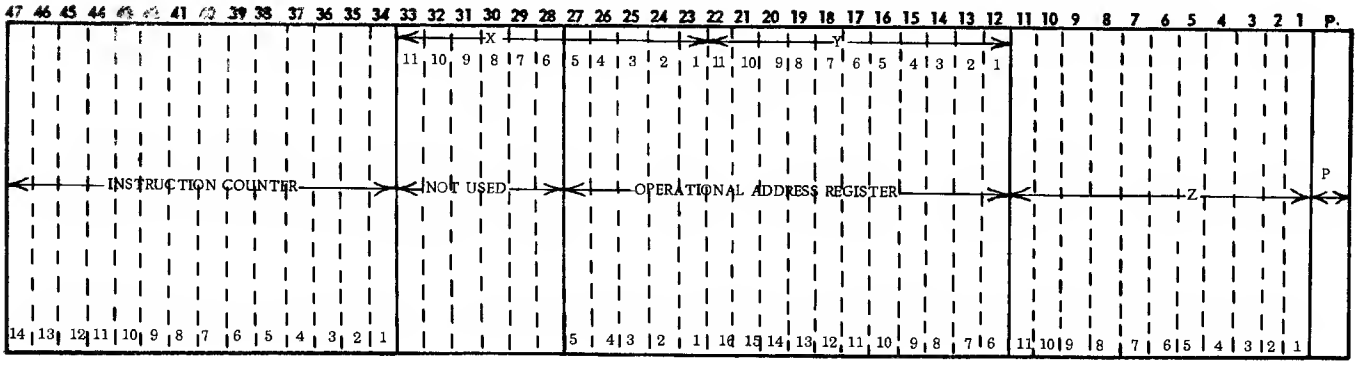
For All Modes Except Copy Mode
For All But Address And Limit Moving Instructions



APPENDIX I - CHART 2

PROCESS WORD

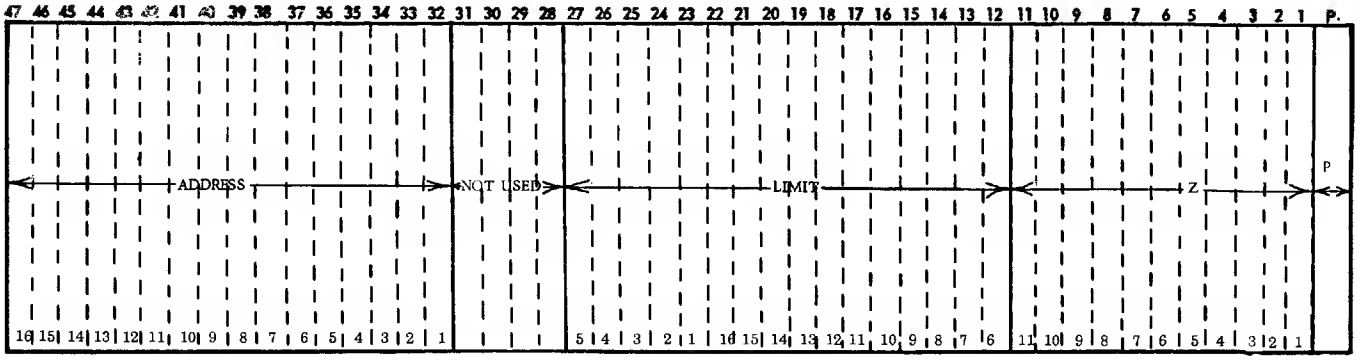
For All Modes Except Copy Mode
For Address And Limit Moving Instructions Only



APPENDIX I - CHART 3

COPY WORD

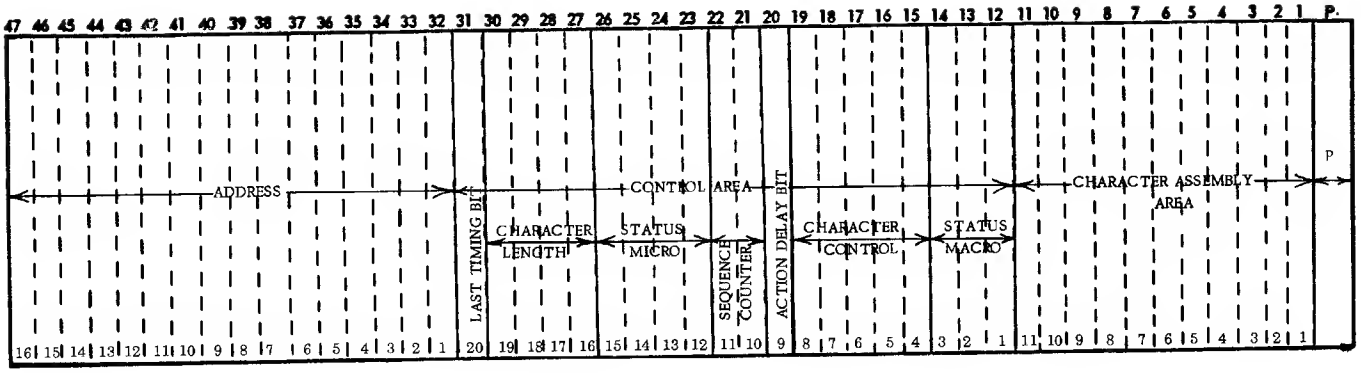
Process Word for Copy Mode



APPENDIX I - CHART 4

CHANNEL WORD

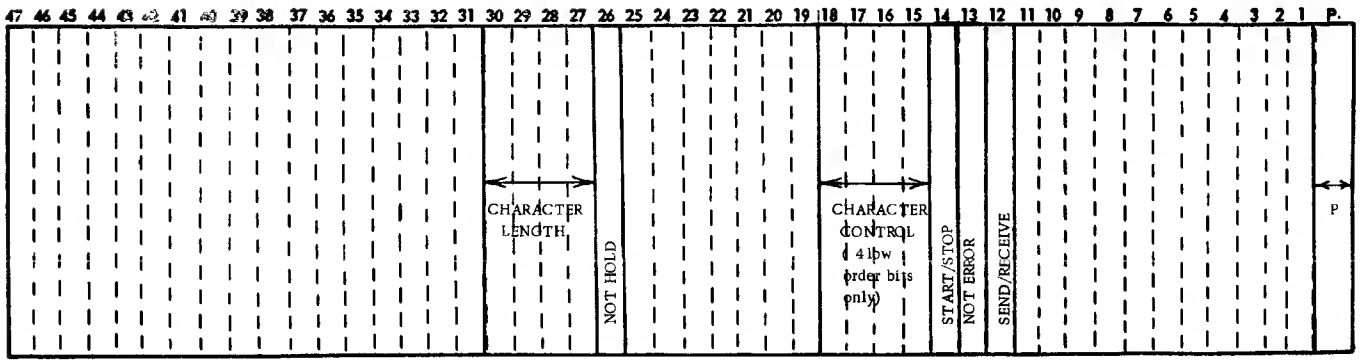
(GENERAL FORMAT)



APPENDIX I - CHART 5

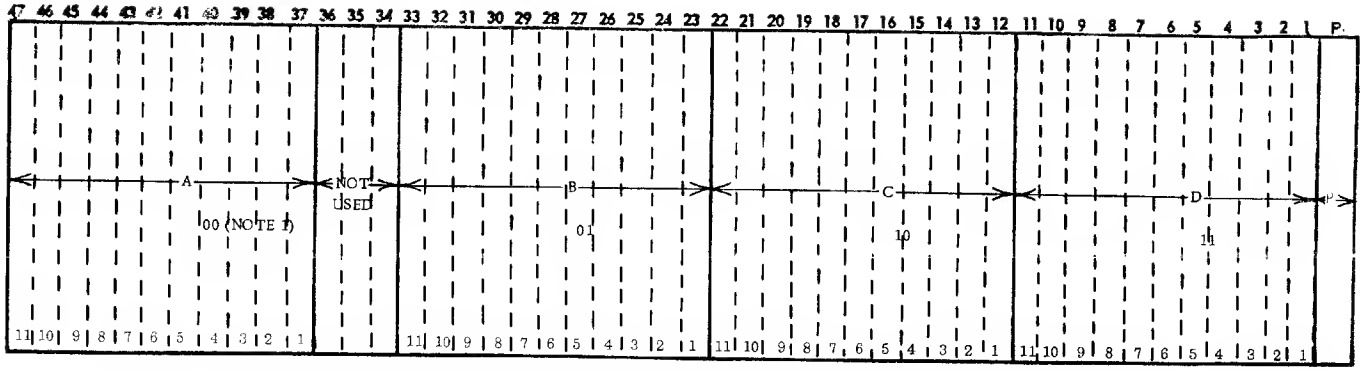
ERROR CHANNEL WORD

(FOR HIGH-SPEED CHANNELS)



APPENDIX II - CHART 1

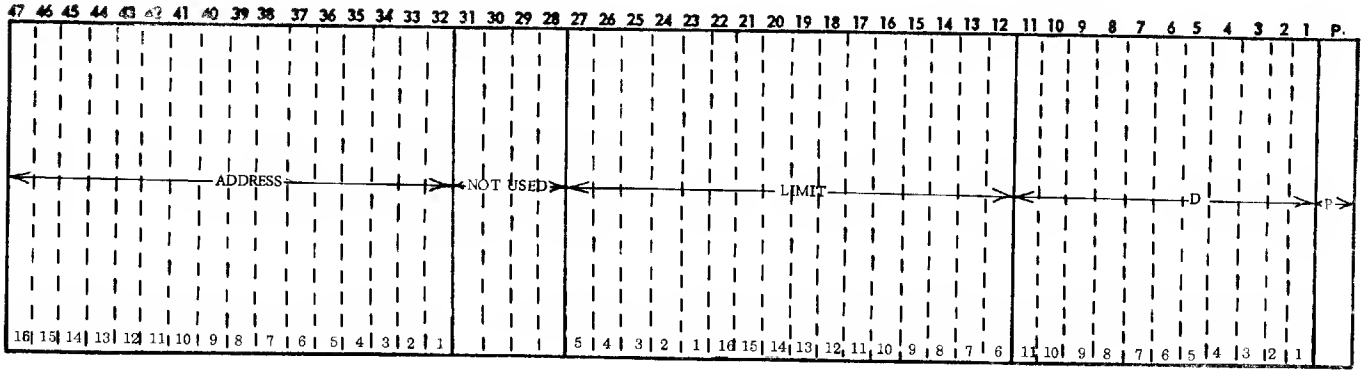
DATA WORD



NOTE 1: THE CHARACTER ADDRESS IS SPECIFIED BY THE TWO LOW ORDER BITS OF A SIXTEEN BIT PROCESS STORAGE ADDRESS.

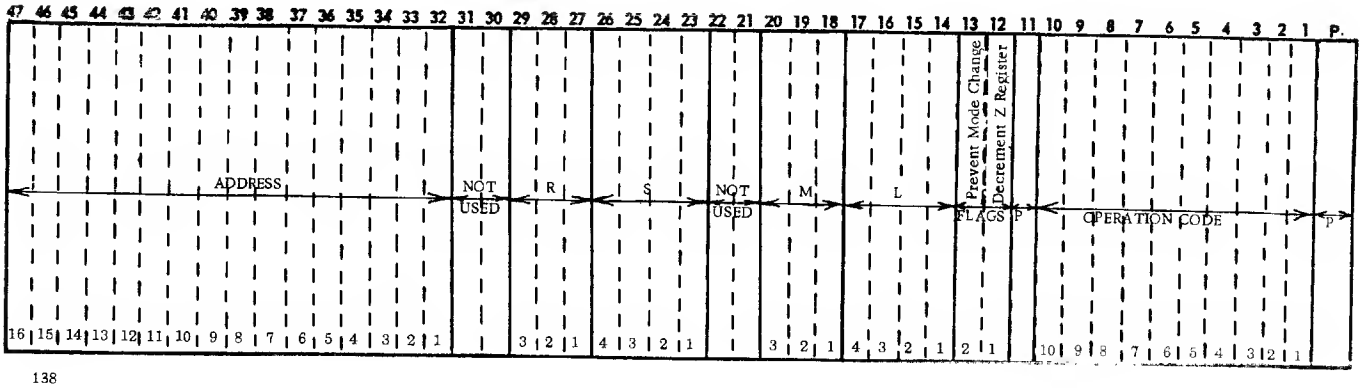
APPENDIX II - CHART 2

LIMIT WORD



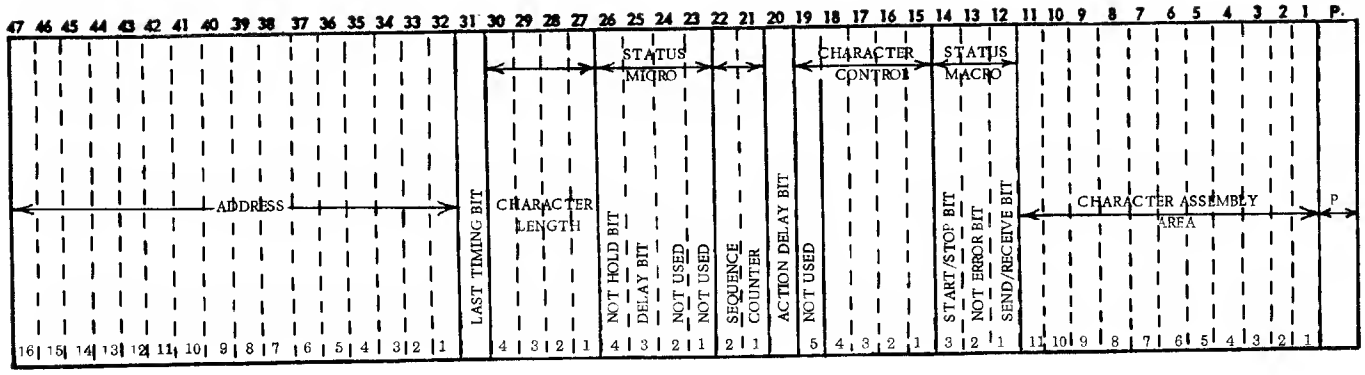
APPENDIX II - CHART 3

INSTRUCTION WORD



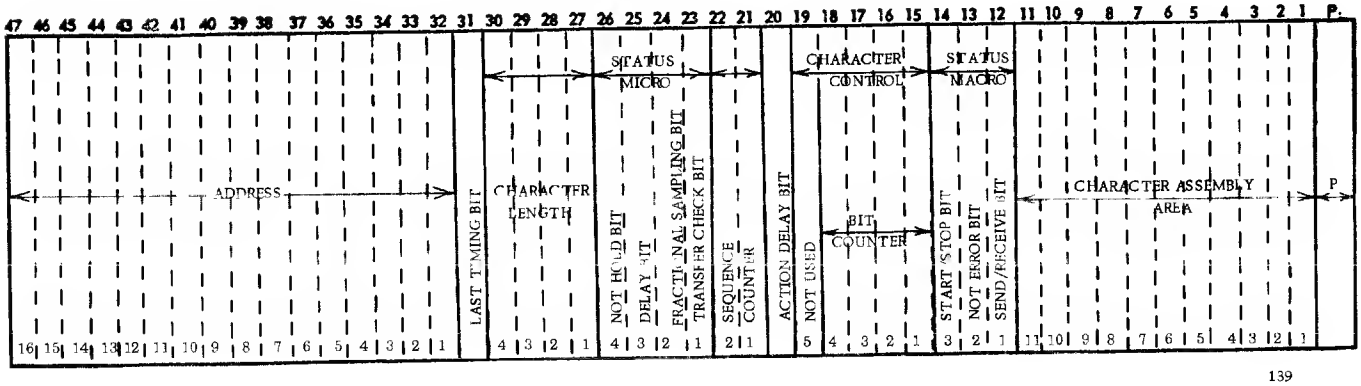
APPENDIX III - CHART 1

START-STOP CHANNEL WORD IN RECEIVE STATUS



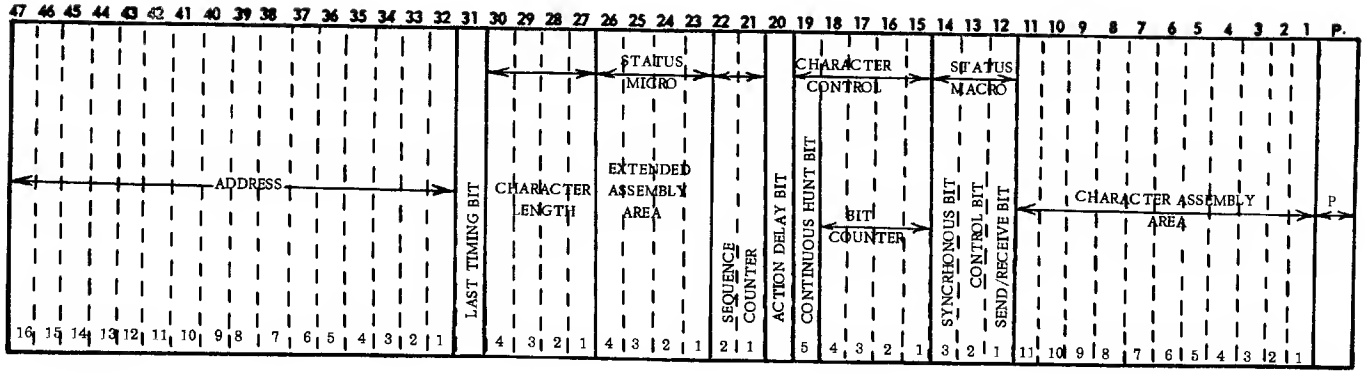
APPENDIX III - CHART 2

START-STOP CHANNEL WORD IN SEND STATUS



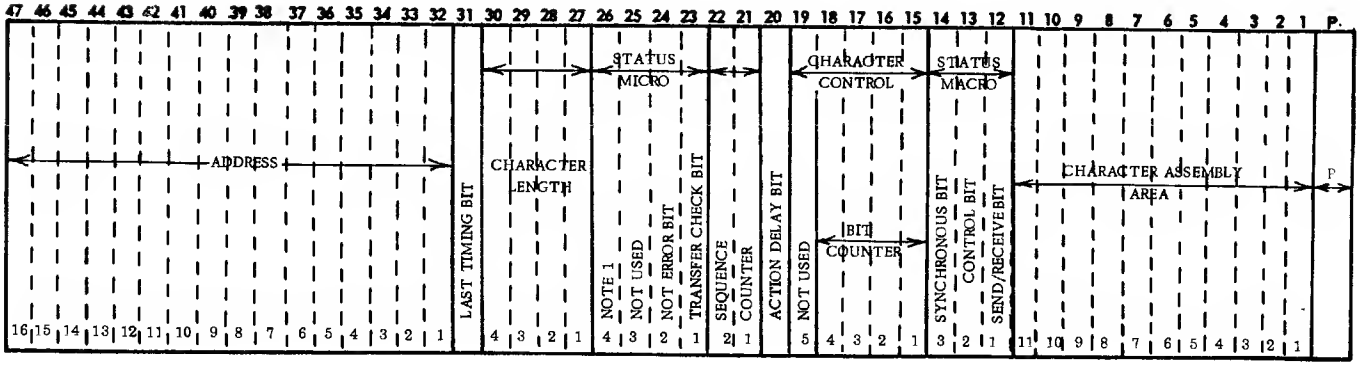
APPENDIX III - CHART 3

SYNCHRONOUS CHANNEL WORD IN RECEIVE STATUS



APPENDIX III - CHART 4

SYNCHRONOUS CHANNEL WORD IN SEND STATUS

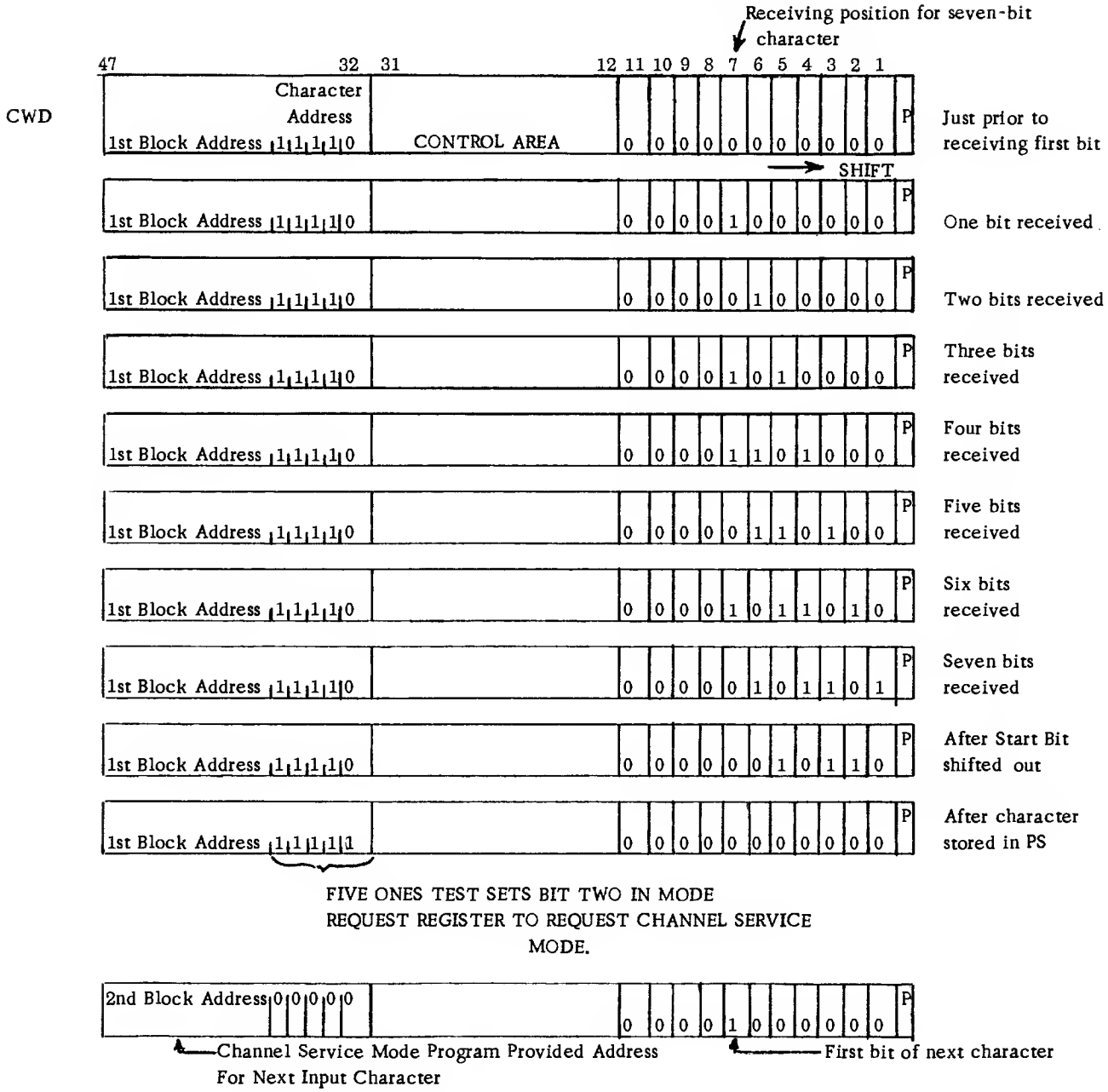


NOTE 1: IT IS USED ONLY WHEN THE CHANNEL IS INACTIVATED AFTER A DATA TRANSFER ERROR.

APPENDIX IV - CHART 1

CHARACTER ASSEMBLY

of seven bit Start-Stop character 0101101



NOTE:
During each Scan Cycle, the Address Line Signals in the Adapter Control Interface are sent to the Control Storage Address Register via the Scanner. This address is then used to read the Channel Word out of Control Storage to the Control Storage Data Register.

APPENDIX IV - CHART 2

PRIORITY PROCESSING MODES

ONE BIT IN DESIRED POSITION IN MODE REQUEST REGISTER:	MODE					
	NORMAL	IN (TO THE 7750)	OUT (FROM THE 7750)	COPY	CHANNEL SERVICE	SERVICE
TURNED ON:	(1)			(2)		
MANUALLY FROM OPERATOR'S PANEL		---	---	---	---	Yes
AUTOMATICALLY, ON OCCURRENCE OF AN ERROR CONDITION		---	---	---	---	Yes
AUTOMATICALLY, WHEN ANY CHANNEL NEEDS IT		---	---	---	Yes	---
BY THE IBM 7750 PROGRAM		Yes	Yes	Yes(3)	Yes(4)	Yes
BY A COMMAND FROM THE ASSOC-IATED COMPUTER		Yes	Yes	---	---	---
TURNED OFF:	(1)					
AUTOMATICALLY, WHEN APPROPRIATE CONDITIONS ARE MET		---	---	Yes	---	---
BY THE IBM 7750 PROGRAM		Yes	Yes	---	Yes	Yes

1.) The 7750 operates in the Normal Mode whenever no other mode is requested (i.e. when the Mode Request Register and the Mode Status Registers contain all Zeros).

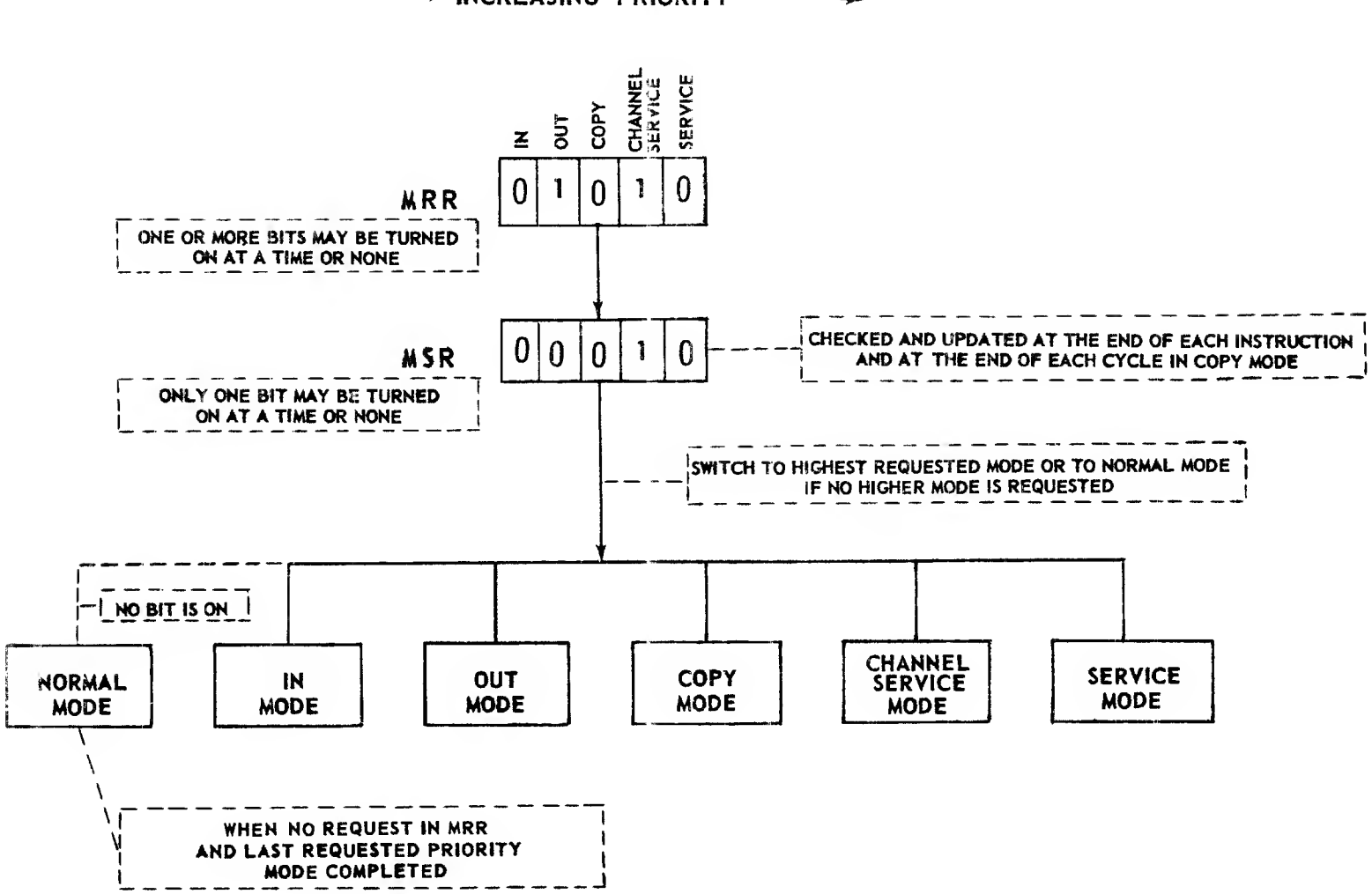
2.) During manually-initiated loading or unloading, the copy mode bit is automatically turned on.

3.) Turning on the copy bit will result in data transfer only if the In or Out bit has previously been set by the associated computer.

4.) The Channel Service Mode should not be requested in this manner because its automatic request is designed to utilize the EOB recognition feature.

APPENDIX IV — CHART 3

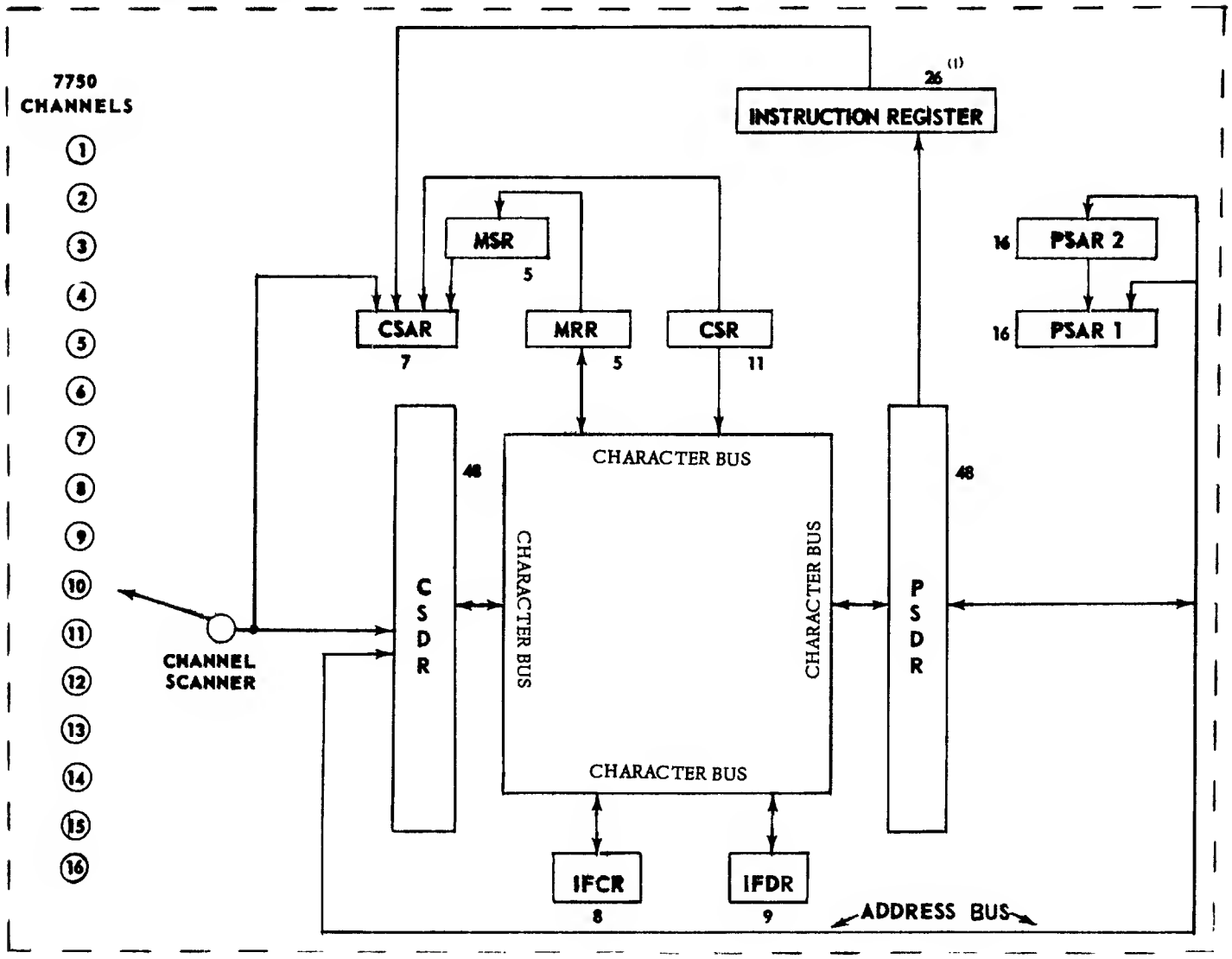
MODE SELECTOR OPERATION



APPENDIX IV — CHART 4

ADDRESSABLE AND NON-ADDRESSABLE IBM 7750 PROCESS CONTROL REGISTERS

RACK B: PROCESS CONTROL

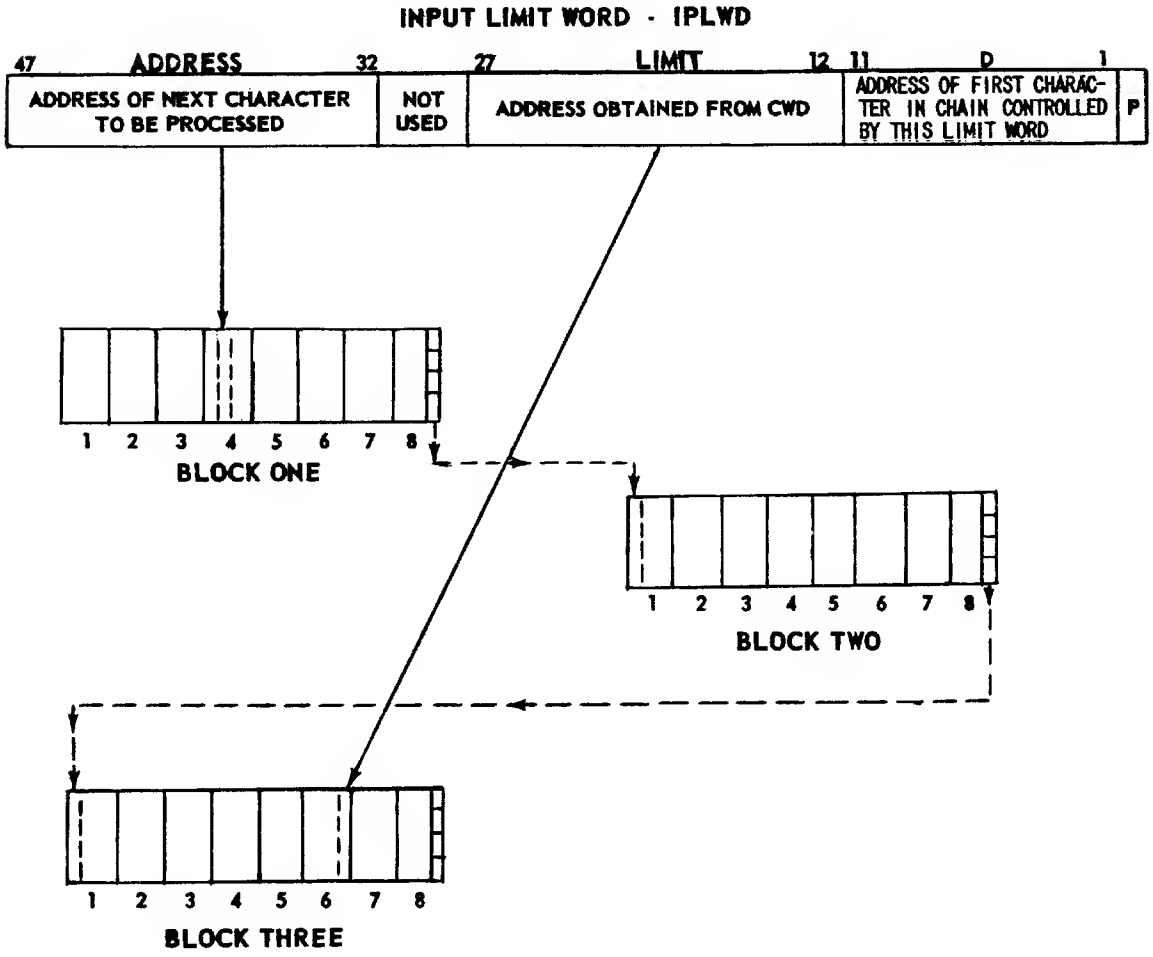


In addition to the addressable registers CSR, IFDR, IFCR and MRR there are 3 other registers, X, Y, and Z located in Process Words and addressable by each mode program.

(1) The numbers outside the blocks represent the number of bits in each register.

APPENDIX IV — CHART 5

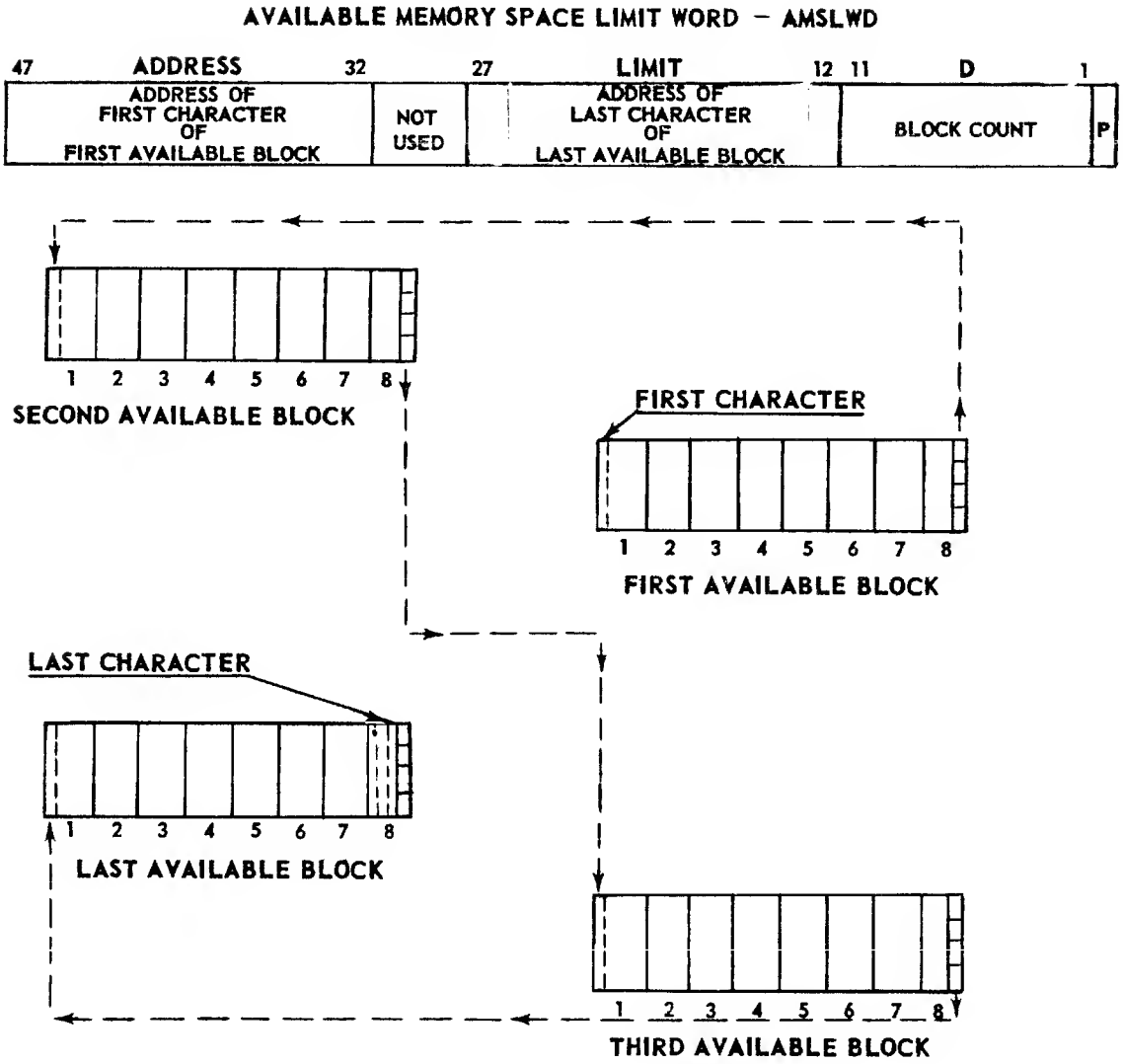
BLOCK CHAINING AND AREA CONTROLLING BY A LIMIT WORD



Block Control Character containing address of next block.

APPENDIX IV — CHART 6

AVAILABLE MEMORY SPACE INVENTORY



Block Control Character containing address of next block.

APPENDIX IV-CHART 7

LIST OF ABBREVIATIONS

ADC	-	ACTION DELAY CHARACTER
CDC	-	CALL DIRECTING CODES
CS	-	CONTROL STORAGE
CSAR	-	CONTROL STORAGE ADDRESS REGISTER
CSDR	-	CONTROL STORAGE DATA REGISTER
CSR	-	CHANNEL SERVICE REGISTER
CWD	-	CHANNEL WORD
DCC	-	DELAY COUNT CHARACTER
DWD	-	DATA WORD
ECWD	-	ERROR CHANNEL WORD
EOB	-	END OF BLOCK CONDITION
HSA1	-	TYPE 1 HIGH-SPEED CHANNEL ADAPTER
HSA2	-	TYPE 2 HIGH-SPEED CHANNEL ADAPTER
IC	-	INSTRUCTION COUNTER
IFCR	-	INTERFACE CONTROL REGISTER
IFDR	-	INTERFACE DATA REGISTER
IR	-	INSTRUCTION REGISTER
IWD	-	INSTRUCTION WORD
LWD	-	LIMIT WORD
MCA	-	MULTIPLEXING CHANNEL ADAPTER
MRR	-	MODE REQUEST REGISTER
MSR	-	MODE STATUS REGISTER
OAR	-	OPERATIONAL ADDRESS REGISTER
PS	-	PROCESS STORAGE
PSAR1	-	PROCESS STORAGE ADDRESS REGISTER 1
PSAR2	-	PROCESS STORAGE ADDRESS REGISTER 2
PSDR	-	PROCESS STORAGE DATA REGISTER
PWD	-	PROCESS WORD
SCC	-	STATUS CHANGE CHARACTER
SDC	-	SENDING DELAY CHARACTER
SWD	-	SCRATCH WORD
u	-	MICRO-SECONDS

IBM 7750 Printed in U.S.A. C22-6695

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York